

Современный
Гуманитарный
Университет

Дистанционное образование

1001.01.01;1

Рабочий учебник

Фамилия, имя, отчество _____

Факультет _____

Номер контракта _____

ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА

ЮНИТА 1

ЧИСЛЕННОЕ РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ И НЕЛИНЕЙНЫХ УРАВНЕНИЙ

Разработано В.Н. Варапаевым, д.ф.-м.н., проф.

Рекомендовано Министерством общего
и профессионального образования
Российской Федерации в качестве
учебного пособия для студентов
высших учебных заведений

КУРС: ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА

Юнита 1. Численное решение систем линейных и нелинейных уравнений.
Юнита 2. Численные методы математического анализа и решения
дифференциальных уравнений.

ЮНИТА 1

Изложены принципы математического моделирования, основные понятия
и методы численного решения систем линейных и нелинейных уравнений, к
которым, как правило, сводятся математические модели.

Для студентов Современного Гуманитарного Университета

Юнита соответствует профессиональной образовательной программе № 1

ОГЛАВЛЕНИЕ

ТЕМАТИЧЕСКИЙ ПЛАН	4
ЛИТЕРАТУРА	5
ТЕМАТИЧЕСКИЙ ОБЗОР	6
Глава 1. Особенности реализации математических моделей на компьютерах	6
1.1. Введение	6
1.2. Этапы решения задачи на ЭВМ	7
1.3. Представление чисел в ЭВМ. Виды погрешностей	10
1.4. Неустойчивые задачи и алгоритмы	14
Глава 2. Системы линейных уравнений. Прямые методы	18
2.1. Некоторые основные понятия	18
2.2. Метод Гаусса	24
2.3. Другие прямые методы	29
2.4. Обусловленность систем линейных уравнений	31
Глава 3. Системы линейных уравнений. Итерационные методы	34
3.1. Общее описание	34
3.1.1. Сущность метода	34
3.1.2. Выбор начального приближения	35
3.1.3. Приведение системы к виду, удобному для итераций	35
3.2. Метод простой итерации	37
3.3. Метод Зейделя	40
3.4. Достаточные условия сходимости методов простой итерации и Зейделя	42
3.5. Сравнительная оценка прямых и итерационных методов	47
Глава 4. Матричные задачи и проблема собственных значений	49
4.1. Нахождение обратной матрицы	49
4.2. Вычисление определителя	52
4.3. Нахождение собственных значений и собственных векторов матриц	54
4.4. Определение наибольшего и наименьшего собственных значений итерационным методом	59
Глава 5. Особенности решения нелинейных задач на примере одного уравнения	63
5.1. Некоторые определения	63
5.2. Численное нахождение корня на ЭВМ	66
5.3. Метод половинного деления и метод хорд	68
5.4. Метод простой итерации	74
5.4.1. Описание метода и условия его сходимости	74
5.4.2. Принцип сжатых отображений	76
5.4.3. Геометрический смысл метода итерации	77
5.4.4. Примеры применения метода простой итерации	79
5.5. Метод Ньютона	80
5.5.1. Общее описание и геометрический смысл	80
5.5.2. Условия сходимости и порядок сходимости	83
5.5.3. Пример "капризного" поведения метода Ньютона	85
5.6. Метод секущих	87
5.7. Сравнительная оценка методов	87
Глава 6. Системы нелинейных уравнений	91
6.1. Постановка задачи и ее особенности	91
6.2. Метод простой итерации	93
6.3. Метод Ньютона и его модификации	97
ПЕРЕЧЕНЬ УМЕНИЙ	104
ТРЕНИНГ УМЕНИЙ	105
ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ	141
ГЛОССАРИЙ*	

* Глоссарий расположен в середине учебного пособия и предназначен для самостоятельного заучивания новых понятий.

ТЕМАТИЧЕСКИЙ ПЛАН

Особенности реализации математических моделей на компьютерах. Математическая модель. Вычислительная математика. Этапы решения задачи на ЭВМ. Представление чисел в ЭВМ. Виды погрешностей. Устойчивые и неустойчивые алгоритмы.

Системы линейных уравнений. Основные понятия. Прямые методы: метод Гаусса и некоторые другие. Обусловленность систем линейных уравнений.

Итерационные методы решения систем линейных уравнений. Методы простой итерации и Зейделя, достаточное условие их сходимости. Ускорение сходимости итерационных методов. Метод релаксации. Сравнительная оценка прямых и итерационных методов. Итерационное уточнение решения.

Матричные задачи. Нахождение обратной матрицы. Вычисление определителя. Полная проблема собственных значений. Определение наибольшего (или наименьшего) собственного значения и соответствующего собственного вектора степенным методом.

Решение одного нелинейного уравнения. Корни уравнения. Сходимость итерационных процессов. Отделение корней. Начальное приближение. Область притяжения корня. Порядок сходимости. Погрешность нахождения корня. Невязки уравнения. Двусторонние приближения. Монотонно сходящиеся и колеблющиеся процессы. Методы половинного деления и хорд. Метод простой итерации. Достаточное условие сходимости. Принцип сжатых отображений. Метод Ньютона. Условия Фурье. Условия и порядок сходимости. Метод секущих. Сравнительная оценка методов. Совместное применение методов хорд и Ньютона. Метод Стеффенсена.

Системы нелинейных уравнений. Связь задачи решения системы нелинейных уравнений и задачи минимизации функции многих переменных. Особенности решения систем нелинейных уравнений. Метод простой итерации и обобщение метода Зейделя. Метод Ньютона. Упрощенный метод Ньютона.

ЛИТЕРАТУРА

Базовая

- *1. Турчак Л. И. Основы численных методов: Учебное пособие. – М.: Наука, 1987.
- 2. Боглаев Ю. П. Вычислительная математика и программирование: Учебное пособие для студентов вузов. – М.: Высшая шк., 1990.

Дополнительная

- *3. Бахвалов Н. С., Жидков Н. П., Кобельков Г. М. Численные методы: Учебное пособие. – М.: Наука, 1987.
- 4. Калиткин Н. Н. Численные методы. – М.: Наука, 1978.
- 5. Тихонов А. Н., Арсенин В. Я. Методы решения некорректных задач. – М.: Наука, 1986
- *6. Форсайт Дж., Малькольм М., Моулер К. Машинные методы математических вычислений. – М.: Мир, 1980.
- 7. Ортега Дж., Рейнболдт В. Итерационные методы решения нелинейных систем уравнений со многими неизвестными. – М.: Мир, 1975.
- *8. Островский А. М. Решение уравнений и систем уравнений. – М.: Изд.иностр.лит., 1963.
- *9. Хемминг Р. В. Численные методы. – М.: Наука, 1968.
- *10. Форсайт Дж. Е., Моулер К. Численное решение систем линейных алгебраических уравнений. – М.: Мир, 1969.

ГЛАВА 1. ОСОБЕННОСТИ РЕАЛИЗАЦИИ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ НА КОМПЬЮТЕРАХ

1.1. Введение

В настоящее время компьютеры стали надежным, а иногда и единственным, средством решения научных, инженерно-технических, экономических, управленческих задач, в которых они традиционно применялись. Наряду с ними появились многочисленные нетрадиционные области приложения, такие как медицина, психология, социология, лингвистика, юриспруденция, искусство и многие другие. Это в большой степени способствовало созданию математических моделей для разнообразных задач, встречающихся в различных областях человеческой деятельности.

Математическая модель – это математическое описание физической, инженерной, экономической и т.п. задач с помощью типичных математических задач, к которым обычно относят: задачи решения алгебраических и трансцендентных уравнений и систем уравнений; дифференциальных (обыкновенных и в частных производных) уравнений, интегральных и смешанных уравнений и дополнительных соотношений, определяющих единственное решение задачи; задачи интерполяции и аппроксимации функций. Конечно, появление новых областей приложения математики приводит к появлению новых отраслей математики, новых постановок задач и разработке методов их решения. Но перечисленный выше типовый набор математических задач является определяющим для подавляющего большинства встречающихся на практике приложений математики.

Пользуясь методами классической математики, можно непосредственно получить решение некоторых из этих задач в виде явных формул или аналитических зависимостей от известных функций, для которых составлены таблицы или номограммы, в виде сходящихся рядов и т.п. Однако существуют, по крайней мере, три соображения, которые делают необходимым применение компьютеров при математическом моделировании. Во-первых, возможность получения решения в явном виде для встречающихся на практике сложных задач является скорее счастливым исключением, чем правилом. Отсюда следует необходимость решать большинство перечисленных математических задач численно, с использованием компьютеров. Во-вторых, полученные методами классической математики аналитические решения часто имеют достаточно громоздкий вид и требуют использования компьютеров для получения конкретных числовых значений. В-третьих, для получения аналитических решений исходную техническую или экономическую задачу приходится упрощать, вводя различные предположения так, чтобы математическая модель допускала аналитическое решение. В этом случае численное решение полной задачи без упрощающих предположений не только позволяет полностью проанализировать исследуемый процесс, но и оценить

* Жирным шрифтом выделены новые понятия, которые необходимо усвоить. Знание этих понятий будет проверяться при тестировании.

влияние на него тех характерных параметров задачи, которыми мы пренебрегли при упрощении математической модели.

При реализации математических моделей на компьютерах существуют специфические особенности, некоторые из которых далее будут рассмотрены. Однако главную из них отметим сейчас. Классическая математика обычно рассматривает задачи с точки зрения доказательства существования и единственности решения и его характерных свойств. При этом, конечно, применяются и конструктивные алгоритмы, указывающие способ построения решения, но они часто имеют вид бесконечных процессов, дающих решение лишь в пределе. Поэтому возникла **вычислительная математика**, которая занимается построением и анализом численных методов, позволяющих получить достаточно точное решение за конечное число шагов, допускающих возможность оценить сделанную ошибку и, быть может, указать приемы эффективного уменьшения этой ошибки. Не следует думать, что вычислительная математика возникла только после появления компьютеров. Потребности практики заставляли ученых заниматься вычислительной математикой задолго до появления компьютеров, но только их появление привело к революционному преобразованию всей науки и техники и особенно вычислительной математики. Следует подчеркнуть, что сказанное выше ни в коем случае не следует понимать как уменьшение роли классической математики, которая является основой научно-технического прогресса. Возможность математического моделирования любых направлений деятельности человека определяет возможность успешного их развития. Вычислительная математика использует все понятия классической математики, ее определения, методологию и технологические приемы, применяя все это к своим конкретным целям. Не имея общих математических знаний, невозможно успешно заниматься математическим моделированием на компьютерах любых научно-технических, экономических и других задач.

1.2. Этапы решения задачи на ЭВМ

При решении любой задачи на электронной вычислительной машине (ЭВМ) можно выделить следующие этапы: Для простоты рассмотрим решение какой-либо физической задачи, хотя аналогично обстоит дело с любым типом задач: технических, экономических, управленческих и других, несмотря на их специфические особенности.

1. Постановка задачи. На этом этапе формулируется физическая задача, которую необходимо решить, выбирается физическая модель процесса. В зависимости от конечной цели решается вопрос о том, какие физические факторы надо учесть, а какими можно пренебречь. На этом этапе чрезвычайно важно учитывать, какие из предположений приведут к существенному упрощению математической модели и без особого ущерба для физической сути явления позволят значительно легче решить задачу. Поэтому хорошее знание математического аппарата, описывающего рассматриваемые физические процессы, является здесь весьма важным.

2. Формулировка математической модели. Здесь физической модели ставится в соответствие ее математическое описание с помощью алгебраиче-

ческих, дифференциальных или интегральных уравнений и соответствующих начальных и граничных условий. Используя методы классической математики, здесь необходимо выяснить насколько правильно поставлена задача, существует ли ее решение и является ли оно единственным. К сожалению, для некоторых сложных математических моделей (особенно нелинейных) ответы на эти вопросы получить не удастся, и при анализе численного моделирования таких процессов приходится использовать интуитивные и эвристические соображения, физическую сущность задачи, опыт решения подобных задач и результаты модельных и натурных физических экспериментов. На этом этапе весьма желательно сформулировать несколько математических моделей различного уровня сложности, которые с различной степенью детализации описывают рассматриваемый физический процесс. Это имеет смысл не только с физической, но и с вычислительной точки зрения, так как независимая реализация этих моделей на ЭВМ позволяет осуществить сравнительный контроль получаемых численных результатов.

3. Выбор численного метода решения задачи. Этот этап заключается в построении приближенного численного метода решения сформулированной математической задачи и формулировке алгоритма реализации этого метода на компьютере. На практике здесь возможен один из трех подходов:

- воспользоваться одним из известных численных методов, разработанных в вычислительной математике для широкого круга математических задач;
- разработать собственный метод решения задачи;
- воспользоваться известным методом, внося в него собственные коррективы, учитывающие специфические особенности решаемой задачи.

В данном учебном пособии в основном излагаются типичные численные методы, необходимые на этом этапе. Отметим, что выбор численного метода и алгоритма его реализации в каждом конкретном случае зависят от типа ЭВМ, на которой планируется решать задачу, и от времени, которым располагаем для ее решения.

4. Составление и отладка программы. Разработанный алгоритм должен быть записан на одном из алгоритмических языков. На этом этапе необходимо продумать, в каком виде компьютеру будет задана исходная информация, какие нужны промежуточные результаты для анализа и контроля, каков должен быть вид и структура результатов решения задачи (таблицы, рисунки, графики, слайды или кинофильмы, отражающие нестационарный характер решения задачи).

Целью отладки программы является выявление ошибок на всех этапах записи алгоритма и программы. Сравнительно просто, с помощью программы-транслятора с алгоритмического языка, обнаруживаются формальные ошибки в записи программы. Смысловые ошибки ищутся путем расчета специально подобранных контрольных вариантов, результаты решения которых известны, и с помощью комплексов отладочных программ, имеющихся в математическом обеспечении. Последние особенно удобны для проверки логической структуры программы. Заключительным этапом отладки обычно является комплексная отладка программы и алгоритма в целом. Ее цель – проверить схему

организации программы в целом; проверить все возможные пути реализации выбранного алгоритма в зависимости от исходных данных задачи; протестировать программу для тех частных случаев физической задачи, для которых либо существует точное решение, либо известны экспериментальные данные исследования этой задачи, либо существуют надежные численные расчеты других исследователей.

5. Проведение расчетов, анализ результатов и уточнение математической модели. Проведение расчетов при различных значениях параметров, характеризующих физическую модель, и анализ полученных результатов позволяет сделать вывод о том, насколько хорошо принятая математическая модель и численная ее реализация соответствуют решаемой физической задаче. На этом этапе возможны корректировка или уточнение, как самой математической модели, так и применяемого численного метода. Здесь важно также выяснить, какую погрешность допускаем мы при численном моделировании физической задачи, и численными экспериментами установить величину этих погрешностей на отдельных этапах решения задачи. Виды и типы погрешностей, возникающих при математическом моделировании на ЭВМ, будут рассмотрены в следующем параграфе.

Описанные выше этапы решения задачи, отражая суть дела, являются в то же самое время достаточно условными. И связано это главным образом с появлением **новых технологий использования компьютеров**. Цель этих технологий – предельно упростить пользователям компьютеров процесс получения решения своих задач. Не рассматривая весь комплекс этих вопросов, отметим некоторые его стороны, относящиеся к численному моделированию.

1. Использование стандартных программ решения типовых математических задач, составленных квалифицированными специалистами.
2. Использование готовых программ решения научно-технических, экономических и других задач.
3. Использование мощных вычислительных комплексов, решающих широкий класс задач целого направления научных, инженерных, проектных, экономических и других исследований.

Если два первых направления в большой степени были характерны для предыдущих этапов развития вычислительной техники, то третий – это качественно новый подход, основными чертами которого являются: дружественный интерфейс для подготовки исходной информации для решаемого класса задач; широкий набор программ, реализующих разнообразные численные методы для решаемых задач; и, наконец, очень большой выбор способов обработки получаемых численных результатов, включая архивирование, графическую обработку, статистический анализ и т.п.

Наконец, отметим еще одно понятие, появившееся в последние годы, – **вычислительный эксперимент** (моделирование реального процесса на компьютере с использованием математической модели, позволяющее полностью или частично заменить натурный или модельный эксперимент). Широкое развитие математического моделирования и резкий рост возможностей современных компьютеров привели к ситуации, когда численное моделирование многих научно-технических задач оказывается более дешевым

и информативно более полным, чем соответствующие натурные и (или) модельные эксперименты. Более того, во многих случаях численное моделирование и вычислительный эксперимент являются единственным способом получения нужной информации. Это в первую очередь относится к экономическим, социологическим и т.п. прогнозам.

1.3. Представление чисел в ЭВМ. Виды погрешностей

В процессе вычислений в ЭВМ числа могут быть представлены в формах с фиксированной точкой и плавающей точкой. В зависимости от того, какой способ представления чисел принят, различают режимы работы ЭВМ с фиксированной точкой и с плавающей точкой.

В режиме с фиксированной точкой заранее устанавливается, какое количество разрядов отводится для целой части числа, а какое – для дробной части. Диапазон представляемых чисел здесь сравнительно невелик. При работе в этом режиме необходимо в алгоритме решения задачи заранее масштабировать все исходные данные, промежуточные и окончательные результаты, чтобы не допустить переполнения разрядной сетки или получения очень маленьких чисел, что может привести к потере точности. Такой режим обычно используется в специализированных компьютерах и в простых и дешевых моделях персональных компьютеров.

В режиме с плавающей точкой действительные числа представляются в виде $X = \pm q \cdot n^p$, где q – мантисса числа, p – порядок числа, а n – основание системы счисления, используемой для записи числа X . Мантисса – это правильная десятичная дробь, порядок p – целое число. Если $1/n \leq q < 1$, то число называется нормализованным (это означает, что в мантиссе после запятой стоит цифра, отличная от нуля). Например, в десятичной системе ($n = 10$) числа $M_1 = 10000$; $M_2 = 0,000056$ в нормализованном виде записываются следующим образом: $M_1 = 0,1 \cdot 10^5$; $M_2 = 0,56 \cdot 10^{-4}$. В компьютере числа с плавающей точкой записываются в двоичной системе счисления. Диапазон представимых в компьютере чисел с плавающей точкой гораздо шире, чем у чисел с фиксированной точкой. Этот диапазон зависит от конкретной модели компьютера и применяемого алгоритмического языка.

Отметим, что ЭВМ оперирует с числами, имеющими конечное число значащих цифр и принадлежащих (по модулю) не всей числовой оси, а некоторому интервалу $M_0 < |x| < M_\infty$, где $M_0 > 0$, $M_\infty < \infty$. M_0 называют машинным нулем, а M_∞ – машинной бесконечностью. Любое полученное в компьютере число, меньшее чем M_0 , считается равным нулю. Число, равное или большее, чем M_∞ , приводит к переполнению разрядной сетки ЭВМ и прекращению вычислений.

На всех этапах математического моделирования на ЭВМ возникают погрешности, которые искажают окончательный результат. Рассмотрим источники этих погрешностей.

(а) **Неустраняемая погрешность**, содержащаяся в математической модели. Она вызвана:

1) неточностью математического описания исследуемого физического явления (не все характерные параметры в нем учтены, так как при переходе

от физической к математической модели всегда используются некоторые упрощающие предположения);

2) неточностью задания величин, входящих в математическую модель (например, коэффициентов и правых частей уравнений, которые могут зависеть от физических характеристик, определяемых экспериментально с некоторой погрешностью). Знание примерной величины неустраняемой погрешности важно прежде всего для того, чтобы оценить требования к точности численного алгоритма решения задачи. Не имеет смысла слишком точно решать задачу, в самой постановке которой уже допущена большая погрешность.

(б) **Погрешность численного метода** обусловлена приближенным способом решения точной математической задачи. В отличие от предыдущей, эта погрешность является регулируемой, так как она может быть уменьшена до любого разумного значения путем изменения одного или нескольких параметров, характеризующих численный метод (например, количества итераций, шага интегрирования, шагов применяемой сетки при переходе от непрерывной модели к дискретной и т.п.).

(в) **Вычислительная погрешность** обусловлена конечной разрядностью представления чисел в компьютере, в результате чего происходит округление результатов при выполнении арифметических операций над числами и при вводе и выводе данных. Последнее вызвано тем, что перевод чисел из десятичной системы счисления, в которой задаются исходные данные, в двоичную форму представления в компьютере происходит приблизительно из-за ограниченной разрядности представления чисел в ЭВМ. Остановимся подробнее на вычислительной погрешности, так как погрешность численных методов будет рассматриваться в последующих главах при изучении самих методов.

Различают два вида погрешностей – абсолютную и относительную. Если X_* – точное значение некоторой величины, а X – известное приближение к нему, то **абсолютной погрешностью** приближения X называют величину $\Delta(X)$, зависящую от X , про которую известно, что

$$|X - X_*| \leq \Delta(X), \quad (1.1)$$

где X_* – точное значение величины X .

Относительной погрешностью $\delta(X)$ для приближения X называют величину, для которой

$$\left| \frac{X - X_*}{X} \right| \leq \delta(X) \quad (1.2)$$

Относительную погрешность иногда выражают в процентах, так как она является долей абсолютной погрешности в значении X .

Следует подчеркнуть, что величины $\Delta(X)$ и $\delta(X)$ являются верхними границами для возможных ошибок величины X , т.е. наибольшими возможными значениями ошибок. Поэтому иногда абсолютными и относительными

погрешностями называют величины $\Delta X = X - X_*$ и $\delta X = \Delta X / X_*$, которые на практике не могут быть использованы, так как истинное значение X_* обычно неизвестно. Определяемые формулами (1.1) и (1.2) погрешности в этом случае называют предельными погрешностями, оценку для которых можно получить, зная погрешности для исходных данных и анализируя их изменение при различных арифметических операциях.

46. Введем еще несколько определений, часто используемых на практике. **Значащими цифрами** числа называют все цифры в его записи, начиная с первой слева ненулевой цифры. Например, в числах 0,005607 и 0,00560700 значащими являются подчеркнутые цифры, число значащих цифр в первом случае равно 4, во втором – 6. **Значащую цифру называют верной**, если абсолютная погрешность числа не превосходит единицы разряда, соответствующего этой цифре. Например, при $X = 0,005607$ и $\Delta(X) = 0,0000009$ все подчеркнутые значащие цифры являются верными. Иногда используется термин **число верных цифр после запятой**. Под ним понимается число цифр после запятой от первой цифры до последней верной цифры. Например, при $X = 0,005607$ это число равно 6.

Очень важным является вопрос о том, как сказываются погрешности округления, возникающие при выполнении каждой арифметической операции, на точность окончательных результатов, особенно если учесть, что количество выполняемых операций для многих сложных задач может достигать величины $10^6 + 10^{12}$ и больше. Несмотря на существование различных подходов (детерминированных и вероятностных), удовлетворительных и надежных теорий в настоящее время нет. Приведем здесь некоторые описанные в литературе соображения, используемые на практике.

Знание абсолютной погрешности позволяет записать информацию о величине в виде возможных пределов ее изменения

$$X - \Delta(X) \leq X_* \leq X + \Delta(X)$$

или иначе

$$X_* = X \pm \Delta(X).$$

Например, запись $X_* = 2,15 \pm 0,01$ означает, что точное значение величины X_* (вообще говоря, неизвестное) лежит в интервале $2,14 \leq X_* \leq 2,16$. На этом основан один из подходов к проблеме округления чисел при решении задачи на компьютере. Каждая величина изображается на самом деле двумя числами – максимальным и минимальным значениями, которые она может принимать. Другими словами, каждое число заменяется областью, в которой лежит точное значение величины (областью ответа). При выполнении операций над величинами вычисляется новая область из исходных областей, используя соответствующие округления. Это позволяет на каждой стадии вычислений определять надежные границы, в которых лежит верный

ответ. При практической реализации (например, с помощью специальных программ) этот метод более чем вдвое увеличивает время вычисления и требует вдвое больше памяти, чем при обычных вычислениях. Тем не менее, в задачах средней сложности такой подход может быть использован. Однако в более сложных задачах такой подход обычно неприменим в силу того, что границы области ответа становятся настолько большими, что говорить о значении ответа становится бессмысленным.

Другим способом оценки влияния округления является решение задачи с обычной и двойной точностью. Принято считать, что совпадающие в двух ответах разряды верны. Выполнив таким образом несколько вариантов типичных вычислений, далее можно вести остальные расчеты с обычной точностью, полагая, что и в них верно то же самое количество разрядов. Хотя опасности этого метода очевидны, тем не менее он используется на практике и обычно дает более точные результаты, чем метод "области ответа".

Существуют и подходы, использующие статистические методы оценки влияния округления. Ознакомиться с ними можно, например, в [9].

Без вывода приведем оценки погрешностей для отдельных арифметических операций над приближенными числами. Для простоты обозначим

$$\Delta X = \Delta(X); \Delta Y = \Delta(Y); \delta X = \delta(X); Y = \delta(Y).$$

При сложении или вычитании чисел их абсолютные погрешности складываются

$$\Delta(X \pm Y) = \Delta X + \Delta Y \quad (1.3)$$

Для относительных погрешностей суммы или разности формулы имеют вид

$$\delta(X + Y) = \frac{\Delta X + \Delta Y}{|X + Y|} = \frac{X}{|X + Y|} \delta X + \frac{Y}{|X + Y|} \delta Y \quad (1.4)$$

$$\delta(X - Y) = \frac{\Delta X + \Delta Y}{|X - Y|} = \frac{X}{|X - Y|} \delta X + \frac{Y}{|X - Y|} \delta Y \quad (1.5)$$

Формулы для погрешностей при умножении и делении имеют вид

$$\Delta(X \cdot Y) = |X| \cdot \Delta Y + |Y| \cdot \Delta X \quad (1.6)$$

$$\Delta(X / Y) = (|Y| \cdot \Delta X + |X| \cdot \Delta Y) / Y^2 \quad (1.7)$$

$$\delta(X \cdot Y) = \delta X + \delta Y \quad (1.8)$$

$$\delta(X / Y) = \delta X + \delta Y \quad (1.9)$$

Из (1.5) следует практически важный вывод о значительной потере точности при вычитании близких чисел. Действительно, при $X \approx Y$ относительная погрешность может быть сколь угодно большой. Пусть, например, $X = 50040$, $Y = 50020$. Абсолютные погрешности исходных данных равны $\Delta X = 0,5$ и $\Delta Y = 0,5$ (половина первого после запятой десятичного разряда, если использовать обычные правила округления). Относительные погрешности исходных данных очень малы $\delta X \approx \delta Y = 0,5/50020 = 0,0001 = (0,01\%)$, а относительная погрешность разности $\delta(X - Y) = [0,5 + 0,5]/20 = 0,05 = (5\%)$. Следовательно, при очень малых относительных погрешностях в исходных данных, мы получили большую погрешность в результате. Значит при организации вычислительных алгоритмов надо стараться видоизменять их так, чтобы по возможности избегать вычитания близких чисел. Приведем пример, когда повышение точности может быть достигнуто за счет несложного алгебраического преобразования.

Пример 1.

Вычислим величину $Y = 70 - \sqrt{4899}$, считая что вычисления производятся в десятичной системе счисления, удерживая после округлений 4 разряда. После округления получаем $\sqrt{4899} = 69,992 \approx 69,99$. Поэтому $Y = 70 - 69,99 = 0,01$. Значение Y можно вычислить иначе, избавившись от иррациональности в числителе

$$Y = \frac{(70 - \sqrt{4899}) \cdot (70 + \sqrt{4899})}{(70 + \sqrt{4899})} = \frac{70^2 - 4899}{(70 + \sqrt{4899})} = \frac{1}{70 + \sqrt{4899}}$$

После округления $70 + \sqrt{4899} = 139,99 \approx 140$, $1/140 = 0,00714285$ и после округления $Y = 0,007143$. Проведя вычисления с большим количеством разрядов, можно проверить, что подчеркнутые в обоих результатах цифры являются верными. Таким образом, в первом случае мы имеем одну верную значащую цифру, во втором – четыре. Потеря значащих цифр здесь произошла при вычитании близких чисел.

Другой практический вывод, следующий из формул для погрешностей, заключается в том, что при вычислении суммы нескольких слагаемых (с округлениями) мы получим меньшую абсолютную погрешность, если будем производить сложение чисел в порядке возрастания их абсолютных величин.

1.4. Неустойчивые задачи и алгоритмы

В математической стране, образно говоря, есть пропасти и ущелья, куда можно свалиться; но это не должно удерживать нас от прогулок.

Л.Ф.Ричардсон

Одной из важных и сложных проблем при численном моделировании на компьютерах математических задач является проблема устойчивости. Обсудим связанные с этим термином понятия.

Рассмотрим неустранимую погрешность математической модели, связанную с наличием погрешности в исходных данных. Необходимо иметь представление о том, как эта погрешность влияет на точность окончательных результатов. Для многих задач погрешность результатов имеет порядок погрешности данных. Такие задачи можно считать "хорошими", поскольку они слабо чувствительны к неточностям в исходных данных. Однако существуют и "плохие" задачи, которые весьма чувствительны к погрешностям в исходных данных. Эта чувствительность характеризуется **устойчивостью**. Задача называется **устойчивой** по исходным величинам, если малые погрешности исходных величин приводят к малым изменениям в окончательных результатах. Задача считается **неустойчивой**, если незначительные погрешности исходных величин приводят к большим погрешностям в результатах или вообще к неверному результату. Решать такие задачи обычными численными методами, как правило, не имеет смысла из-за невозможности получить правильное решение. Однако в настоящее время разработаны способы решения некоторых из этих задач, основанные на специальных методах регуляризации. Ведущая роль в разработке этих методов принадлежит советской математической школе (см. например [5]).

Приведем простой пример неустойчивой задачи. Пусть надо найти действительные корни многочлена $(X - a)^n = \varepsilon$, где ε – малое число ($\varepsilon > 0$). Видно, что изменение правой части на величину порядка ε приводит к погрешности вычисления корней порядка $1/n$. Например, при $n=10$ погрешность правой части порядка 10^{-10} приведет к ошибке в нахождении корня $\sim 0,1$.

Однако существуют не только "хорошие" и "плохие" задачи, но и "хорошие" и "плохие" алгоритмы решения задач. Решая "хорошую" задачу с помощью "плохого" алгоритма, ее решение получить не удастся.

Устойчивым называется алгоритм, в котором малые ошибки в начальных данных и возникающие в процессе счета погрешности округления убывают либо умеренно возрастают, не приводя к существенному искажению окончательных результатов. Для **неустойчивых алгоритмов** погрешности могут нарастать настолько быстро, что уже за небольшое количество операций будет получен совершенно неверный результат. Для иллюстрации сказанного приведем простой пример неустойчивого алгоритма.

Пример 2.

Требуется вычислить значение интеграла

$$J_n = \frac{1}{e} \int_0^1 x^n e^x dx \quad (1.10)$$

при различных значениях параметра $n = 0, 1, 2, \dots, 20$, где e – основание натурального логарифма ($e = 2,71\dots$).

Для решения этой задачи, конечно, можно было бы воспользоваться имеющимися на каждой ЭВМ стандартными программами вычисления определенных интегралов. Но в данном случае можно значительно упростить задачу, сведя ее решение к выполнению небольшого числа элементарных операций. Проинтегрировав (1.10) по частям, легко получить соотношение

$$J_n = 1 - n \cdot J_{n-1}, \quad (1.11)$$

связывающее значения интеграла при двух последовательных значениях параметра n (такие соотношения называются в математике рекуррентными). При $n = 0$ интеграл (1.10) легко вычисляется

$$J_0 = \frac{1}{e} \int_0^1 e^x dx = \frac{1}{e} (e - 1) = 1 - \frac{1}{e}. \quad (1.12)$$

Имея начальное значение J_0 , мы можем с помощью соотношения (1.11) последовательно определять значения величин $J_1, J_2, \dots, J_{20}$. При этом для получения каждого очередного значения J_n требуется всего две операции (умножение и вычитание). Для решения поставленной задачи всего требуется $2 \cdot 20 = 40$ операций. Математически можно показать, что последовательность J_n ограничена и стремится к нулю, т.е.

$$0 < J_{n+1} < J_n, \quad J_n \rightarrow 0 \quad (n \rightarrow \infty) \quad (1.13)$$

На первый взгляд может показаться, что этот алгоритм прост и легко решает задачу. Однако проведение расчетов на компьютере приводит к парадоксальным результатам. В таблице 1.1 приведены результаты, полученные на ЭВМ. Поскольку исходная величина J_0 зависит от иррационального числа e , беря различное количество значащих цифр в записи числа e , мы получаем различные значения погрешности исходной величины J_0 .

Таблица 1.1

n	0	2	4	6	7	8
$J_n(\text{точн.})$	0,63212	0,26414	0,170893	0,126802	0,112384	0,100932
J_n	0,63212	0,26414	0,170887	0,126624	0,113631	0,090955

n	9	10	11	12	13	14
$J_n(\text{точн.})$	0,091612	0,083877	0,077352	0,071773	0,0669477	0,0627322
J_n	0,181409	-0,81409	9,95497	-118,460	1540,975	-21572,66

В таблице приведены результаты J_n для $e=2,71828$. Из таблицы видно, что, начиная с $n \approx 9$, результаты становятся совершенно неверными, так как они не удовлетворяют аккуратным математическим свойствам (1.13) для последовательности J_n . Причем проведение расчетов с большей точностью может лишь несколько отодвинуть границу значений n , при которых происходит стремительное нарастание ошибки, не изменяя неустойчивости алгоритма. Так, при задании значения e с 10-ю знаками устойчивость теряется при $n \approx 16$.

Чтобы разобраться в причинах неустойчивости алгоритма (1.11), рассмотрим, как изменяется начальная ошибка в процессе счета. Приведенные ниже рассуждения не являются строгим математическим доказательством, но вполне характеризуют суть происходящего. Пусть $\overline{J_n}$ – есть точное решение

задачи (1.11) с начальным условием (1.12) в смысле классической математики, то есть содержащее в своей записи бесконечное количество разрядов, а ε_n – погрешность, возникающая на n -ом шаге алгоритма в реальных вычислениях на компьютере. Тогда результат решения задачи на компьютере можно представить в виде

$$J_n = \overline{J_n} + \varepsilon_n. \quad (1.14)$$

Рассмотрим, как изменяется погрешность на одном шаге алгоритма.

Подставим (1.14) в (1.11) и учтем, что для точного решения $\overline{J_n}$ соотношение

$$(1.11) \text{ справедливо, т.е. } \overline{J_n} = 1 - n\overline{J_{n-1}}.$$

Тогда $\overline{J_n} + \varepsilon_n = 1 - n(\overline{J_{n-1}} + \varepsilon_{n-1})$ и, следовательно

$$\varepsilon_n = -n \cdot \varepsilon_{n-1} \quad (1.15)$$

Если в начальном выражении (1.12) допущена ошибка ε_0 (например, за счет неточно заданного числа e , в записи которого мы удерживаем конечное число разрядов), то, последовательно применяя (1.15), получим, что к n -ому шагу эта погрешность изменится следующим образом

$$\varepsilon_n = (-1)^n \cdot n! \cdot \varepsilon_0 \quad (1.16)$$

Формула (1.16) показывает, что с ростом n погрешность нарастает чрезвычайно быстро, хотя ε_0 может быть и очень малым, например, порядка 10^{-8} . Видно также, почему в таблице 1, начиная с некоторых n , знаки у величины J_n изменяются на каждом шаге, несмотря на наличие точного теоретического соотношения (1.13). Дело в том, что, начиная с некоторых n , погрешность решения ε_n “забывает” основное решение $\overline{J_n}$ и $J_n \approx \varepsilon_n$, так как ε_n растет, а $\overline{J_n}$ убывает. На практике использование таких неустойчивых алгоритмов недопустимо.

Интересно отметить, что, несколько видоизменив алгоритм, можно сделать его устойчивым. Для этого, воспользовавшись тем, что $\overline{J_n} \rightarrow 0$ при $n \rightarrow \infty$, положим J_n равным нулю при каком-либо большом n , например,

$$J_{1000} = 0. \quad (1.17)$$

Выразим теперь J_{n-1} через J_n с помощью соотношения (1.11)

$$J_{n-1} = \frac{1}{n}(1 - J_n) \quad (1.18)$$

(1.18) позволяет, начиная, например, с J_{1000} , последовательно определить все J_n с меньшими номерами. Легко проверить, что в отличие от (1.11) алгоритм,

основанный на (1.18), будет устойчив, что позволяет полностью решить на ЭВМ поставленную в рассматриваемом примере задачу. Для (1.18) погрешность будет быстро убывать

$$\varepsilon_{n-1} \sim \frac{(-1)^n \cdot \varepsilon_{1000}}{n!}$$

Приведенный пример показывает, что для решения одной и той же задачи могут быть как устойчивые, так и неустойчивые алгоритмы. Довольно часто вопрос об устойчивости алгоритмов не удается исследовать теоретически и приходится прибегать к численным экспериментам. Заметим также, что существуют алгоритмы, в которых погрешность не нарастает так катастрофически, как в примере 2 (функция $n!$ растет чрезвычайно быстро!). Довольно часто в неустойчивых алгоритмах погрешность нарастает экспоненциально (e^{ax} – тоже быстро растет). Более того, иногда считают, что если погрешность округления нарастает по степенному закону, то алгоритм считают условно устойчивым (устойчивым при некоторых ограничениях на объем вычислений и требуемую точность). Смысл такого подхода основан на том, что погрешности округления искажают последние значащие цифры и, следовательно, вычисления надо закончить до того, как “волна погрешности округления” дойдет до первых значащих цифр.

86 В заключение приведем еще одно часто используемое понятие. **Задача называется поставленной корректно**, если ее решение существует при любых допустимых входных данных, решение является единственным и оно устойчиво по исходным величинам (малому изменению входных данных соответствует малое изменение решения). **Задача поставлена некорректно**, если ее решение неустойчиво относительно входных данных.

ГЛАВА 2. СИСТЕМЫ ЛИНЕЙНЫХ УРАВНЕНИЙ. ПРЯМЫЕ МЕТОДЫ

2.1. Некоторые основные понятия

610 К основным задачам линейной алгебры обычно относят решение систем линейных алгебраических уравнений, нахождение обратной матрицы и проблему собственных значений матриц. В этой и двух последующих главах мы будем рассматривать именно эти задачи.

610 Системы линейных уравнений и численные методы, применяемые при их решении, образуют фундамент всей вычислительной математики) поскольку:

- эти задачи практически встречаются чаще всего в научных, инженерных, экономических расчетах и в других приложениях;

- к ним сводится, как правило, численное решение более сложных математических задач, таких, например, как решение дифференциальных уравнений (как обыкновенных, так и в частных производных);

- при построении численных методов для нелинейных задач часто конструируются алгоритмы, основанные на последовательном решении линейных задач (например, метод Ньютона для решения системы нелинейных уравнений сводит задачу к последовательному решению систем линейных уравнений).

Система n линейных алгебраических уравнений с n неизвестными в общем виде может быть записана следующим образом

[illegible]

$$Ax = \bar{b}, \quad (2.2)$$
$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \quad \bar{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \bar{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Различные методы решения системы (2.1) заключаются в том, что система (2.1) преобразуется в какой-либо другой вид, откуда решение может быть легко получено. Конечно, преобразования при этом должны быть такими, чтобы они не изменяли решения системы. Преобразования линейной системы уравнений, не изменяющие решение системы и, быть может, упрощающие вид ее матрицы,

называются эквивалентными преобразованиями. Оказывается, общий вид таких преобразований заключается в том, что обе части уравнения системы в матричном виде (2.2) умножаются на одну и ту же невырожденную матрицу (определитель которой не равен нулю) такой же квадратной структуры, как и A

$$BA\bar{x} = B\bar{b}$$

Вводя матрицу $C = AB$, равную произведению матриц A и B и новый вектор $\bar{q} = B\bar{b}$, получим исходную систему (2.2) в преобразованном эквивалентном виде

$$C\bar{x} = \bar{q}. \quad (2.3)$$

Преобразования имеют смысл, если полученную систему решить гораздо легче, чем исходную систему (2.2). Во многих случаях наряду с матрицей A системы удобно рассматривать расширенную матрицу, которая получается добавлением к основной матрице вектора правых частей системы

$$A_p = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{bmatrix}. \quad (2.4)$$

В теории линейной алгебры формулируются условия существования решения для линейной системы (2.1). Система (2.1) имеет решение тогда и только тогда, когда ранг матрицы A и ранг расширенной матрицы (2.4) совпадают. Если при этом ранг матрицы A равен n (это означает, что определитель матрицы A не равен нулю и система является неоднородной), то система (2.1) имеет единственное решение, определяемое по правилу Крамера

$$x_i = D_i / D, \quad (2.5)$$

где D – определитель матрицы A , а D_i – определитель матрицы, полученной из A заменой i -го столбца на столбец правых частей $\bar{b}$.

Однако простота формулировки способа нахождения решения является обманчивой. Все существенно осложняется при переходе к численному решению линейных систем на ЭВМ. Во-первых, применение правила Крамера при больших n на ЭВМ абсолютно исключается, поскольку:

- а) для вычисления одного определителя путем вычисления всех входящих в сумму членов требуется около $n \cdot n!$ арифметических операций; уже при $n > 10$ время для его вычисления становится очень большим и чрезвычайно быстро нарастает с ростом n ;

- b) из-за выполнения очень большого числа операций будут быстро нарастать погрешности округления;
- c) в процессе вычисления определителя придется перемножать большое количество чисел, что может привести к переполнению разрядной сетки компьютера.

Во-вторых, на практике обычно приходится решать системы очень высокого порядка (от нескольких сотен до сотен тысяч уравнений), что приводит к очень большому росту погрешности, если не использовать специальных приемов в процессе решения задачи. В-третьих, при больших n часто сами задачи оказываются "плохими" в том смысле, что малым изменениям в исходных данных соответствуют достаточно большие ошибки в решении системы. Такие системы называются плохо обусловленными, и они будут рассмотрены ниже в параграфе 2.4.

Очень важным является понятие обратной матрицы. Матрицей, обратной матрице A , называется матрица A^{-1} , умножив которую на матрицу A мы получим единичную матрицу

$$AA^{-1} = E \quad (2.6)$$

Зная обратную матрицу A^{-1} , мы можем легко получить решение линейной системы (2.2). Действительно, умножив обе части (2.2) слева на A^{-1} , получим

$$A^{-1}A\bar{x} = A^{-1}\bar{b} \Rightarrow E\bar{x} = A^{-1}\bar{b} \Rightarrow \bar{x} = A^{-1}\bar{b}$$

Следовательно, умножив матрицу A^{-1} на вектор правых частей $\bar{b}$ мы получаем решение системы (2.2). Вообще говоря, нахождение обратной матрицы – задача более сложная, чем решение системы (2.2). Находить ее имеет смысл в случае, когда она представляет самостоятельный интерес для решаемой задачи, либо при необходимости находить ряд решений системы (2.2) с различными правыми частями. Численный метод нахождения обратной матрицы будет рассмотрен в главе 4.

Методы решения систем линейных уравнений можно разделить на две группы – прямые (или точные) и итерационные. **Прямые методы** – это методы, которые при отсутствии погрешностей округления за конечное число арифметических и логических операций дают точный результат. Эти методы сравнительно просты. Их иногда называют точными, поскольку решение выражается в виде точных формул через коэффициенты системы и правые части. Однако точным решение может быть лишь теоретически, при выполнении вычислений с бесконечным числом разрядов. На практике при использовании ЭВМ вычисления проводятся с ограниченным числом разрядов и поэтому неизбежны погрешности в окончательных результатах решения задачи.

Итерационные методы заключаются в построении последовательности приближенных решений, предел которой (если он существует) является решением системы. Эти методы будут подробно разобраны в следующей главе.

Выбор конкретного численного метода определяется специфическими особенностями решаемой задачи и располагаемой вычислительной техникой. В первую очередь здесь необходимо учитывать вид матрицы решаемой линейной системы. Приведем наиболее типичные виды матриц, которые

встречаются в разнообразных практических задачах и при конструировании численных методов.

142 **Диагональная матрица** – квадратная матрица, у которой $a_{ij} = 0$ при $i \neq j$ ($i = 1, 2, \dots, n; j = 1, 2, \dots, n$). Другими словами, у диагональной матрицы ненулевые элементы стоят только на главной диагонали.

Единичная матрица – диагональная матрица, у которой на главной диагонали стоят единицы, т.е. $a_{ii} = 1$ ($i = 1, 2, \dots, n$), $a_{ij} = 0$ при $i \neq j$.

Верхняя треугольная матрица – матрица, у которой ниже главной диагонали все элементы равны нулю, т.е. $a_{ij} = 0$ при $i > j$.

Нижняя треугольная матрица – матрица, у которой выше главной диагонали все элементы равны нулю, т.е. $a_{ij} = 0$ при $i < j$.

Симметричная матрица – квадратная матрица, у которой все элементы, симметричные относительно главной диагонали, равны, т.е.

$$a_{ij} = a_{ji} \quad (i = 1, 2, \dots, n; j = 1, 2, \dots, n).$$

Ленточная матрица – квадратная матрица, у которой все ненулевые элементы образуют “ленту”, параллельную главной диагонали, а все остальные элементы матрицы равны нулю, т.е. $a_{ij} = 0$ при $|i - j| \geq l$, где l – ширина ленты.

Трехдиагональная матрица – квадратная матрица, у которой не равны нулю только те элементы, которые расположены на главной диагонали и двух соседних с ней диагоналях, т.е. $a_{ij} \neq 0$ при $|i - j| \leq 1$. Трехдиагональная матрица – частный случай ленточной матрицы.

На рис.2.1 приведены примеры рассмотренных матриц.

Еще один важный случай – **разреженные матрицы**. Так называются матрицы, у которых количество ненулевых элементов гораздо меньше общего количества элементов. Другими словами, подавляющее большинство элементов разреженной матрицы равны нулю.

136 Рассматривают еще два типа матриц, которые отличаются тем, нужно или нет хранить элементы матрицы в памяти компьютера. Матрица называется **хранимой**, если при проведении расчетов ее элементы необходимо хранить в памяти компьютера. Матрица называется **воспроизводимой** (вычисляемой), если ее элементы не требуется хранить в памяти, так как в процессе вычислений по мере необходимости элементы матрицы могут быть вычислены по простым формулам (воспроизведены). Примером воспроизводимой матрицы является следующая трехдиагональная матрица

$$A = \begin{bmatrix} 4 & 1 & 0 & 0 & 0 \\ -1 & 4 & 1 & 0 & 0 \\ 0 & -1 & 4 & 1 & 0 \\ 0 & 0 & -1 & 4 & 1 \\ 0 & 0 & 0 & -1 & 4 \end{bmatrix}$$

Ее элементы легко вычисляются: $a_{ii} = 4$ на главной диагонали, т.е. при $i = j$; $a_{ij} = 1$ при $j = i + 1$ (выше главной диагонали); $a_{ij} = -1$ при $j = i - 1$ (ниже главной диагонали); $a_{ij} = 0$ в остальных случаях.

$$\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & -7 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}$$

диагональная матрица

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

единичная матрица

$$\begin{bmatrix} 4 & 3 & 1 & 2 \\ 0 & 7 & 6 & -1 \\ 0 & 0 & 3 & -5 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

верхняя треугольная матрица

$$\begin{bmatrix} 9 & 0 & 0 & 0 \\ 3 & 2 & 0 & 0 \\ 2 & 2 & -6 & 0 \\ -7 & 6 & 8 & 4 \end{bmatrix}$$

нижняя треугольная матрица

$$\begin{bmatrix} 2 & -2 & 0 & 9 \\ -2 & 8 & 7 & -6 \\ 0 & 7 & 5 & 4 \\ 9 & -6 & 4 & 1 \end{bmatrix}$$

симметричная матрица

$$\begin{bmatrix} 4 & 2 & 0 & 0 \\ 3 & 1 & 5 & 0 \\ 0 & 4 & 2 & 3 \\ 0 & 0 & 7 & 9 \end{bmatrix}$$

трехдиагональная матрица

$$\begin{bmatrix} 4 & 1 & 2 & 0 & 0 & 0 \\ 3 & 4 & 2 & 1 & 0 & 0 \\ 6 & 1 & 3 & 2 & 5 & 0 \\ 0 & 1 & 2 & 7 & 2 & 4 \\ 0 & 0 & 9 & 3 & 3 & 2 \\ 0 & 0 & 0 & 7 & 6 & 9 \end{bmatrix}$$

ленточная матрица

Рис 2.1 Примеры различных матриц

2.2. Метод Гаусса

Метод Гаусса – прямой метод решения системы линейных уравнений, основанный на сведении исходной системы к треугольному виду с помощью эквивалентных преобразований и последующего решения полученной треугольной линейной системы. Является одним из наиболее эффективных прямых методов решения систем линейных уравнений. Идея метода Гаусса (часто его называют методом последовательного исключения искоемых переменных) заключается в том, чтобы исходную линейную систему $A\bar{x} = \bar{b}$ с помощью эквивалентных преобразований привести к виду

$$U\bar{x} = \bar{b}^*, \quad (2.7)$$

$$\begin{cases} 2x_1 + x_2 - x_3 = 3 \\ 0 - \frac{3}{2}x_2 + \frac{5}{2}x_3 = -\frac{1}{2} \\ 0 + \frac{1}{2}x_2 + \frac{5}{2}x_3 = \frac{7}{2} \end{cases} \quad (2.10)$$

Для того, чтобы привести систему к треугольному виду, нам осталось исключить x_2 из третьего уравнения (2.10). Для этого из третьего уравнения вычтем второе уравнение, умноженное на $-\frac{1}{3}$. В результате получим

$$\begin{cases} 2x_1 + x_2 - x_3 = 3 \\ 0 - \frac{3}{2}x_2 + \frac{5}{2}x_3 = -\frac{1}{2} \\ 0 + 0 + \frac{10}{3}x_3 = \frac{10}{3} \end{cases} \quad (2.11)$$

На этом заканчивается прямой ход метода Гаусса, так как матрица (2.11) является верхней треугольной. Осталось найти решение с помощью обратного хода. Из третьего уравнения (2.11) сразу находим $x_3 = 1$. Подставив это значение во второе уравнение, найдем, что $-\frac{3}{2}x_2 + \frac{5}{2} = -\frac{1}{2}$, откуда $x_2 = 2$. Подставляя $x_3 = 1$ и $x_2 = 2$ в первое уравнение, определим из него значение x_1 : $2x_1 + 2 - 1 = 3$, откуда $x_1 = 1$. Итак, мы получили решение системы (2.9):

$$x_1 = 1, x_2 = 2, x_3 = 1.$$

Рассмотрим теперь прямой ход в общем случае для системы (2.1). Во всех уравнениях, кроме первого, исключается переменная x_1 . Для этого из второго уравнения вычтем первое, умноженное на a_{21}/a_{11} , из третьего – первое, умноженное на a_{31}/a_{11} , вообще, из i -го уравнения вычтем первое, умноженное на a_{i1}/a_{11} . В результате исходная система сведется к виду

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ \quad a_{22}^{(1)}x_2 + \quad + a_{2n}^{(1)}x_n = b_2^{(1)} \\ \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ \quad a_{n2}^{(1)}x_2 + \quad + a_{nn}^{(1)}x_n = b_n^{(1)} \end{cases} \quad (2.12)$$

Верхний индекс коэффициентов и правых частей во всех уравнениях, начиная со второго, указывает на то, что эти значения коэффициентов новые, полученные в результате преобразования системы (2.1). При преобразовании системы мы предполагали, что $a_{11} \neq 0$, так как происходит деление на этот

элемент, называемый ведущим на данном шаге преобразования системы.* Следующим шагом мы исключим переменную x_2 из всех уравнений, кроме первого и второго. Для этого, выбирая в качестве ведущего элемента $a_{22}^{(1)}$, вычтем из третьего уравнения (2.12) второе уравнение, умноженное на $a_{32}^{(1)}/a_{22}^{(1)}$, из четвертого – второе, умноженное на $a_{42}^{(1)}/a_{22}^{(1)}$, вообще, из i -го уравнения вычтем второе уравнение, умноженное на $a_{i2}^{(1)}/a_{22}^{(1)}$ (для всех $i, 3 \leq i \leq n$).

Тогда система (2.12) примет вид

$$\left\{ \begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & a_{13}x_3 & + & & + & a_{1n}x_n & = & b_1 \\ & & a_{22}^{(1)}x_2 & + & a_{23}^{(1)}x_3 & + & \dots & + & a_{2n}^{(1)}x_n & = & b_2^{(1)} \\ & & & & a_{33}^{(2)} & + & \dots & + & a_{3n}^{(2)} & = & b_3^{(2)} \\ & & & & \dots & & & & \dots & & \dots \\ & & & & a_{n3}^{(2)} & + & \dots & + & a_{nn}^{(2)} & = & b_n^{(2)} \end{array} \right. \quad (2.13)$$

На следующем шаге, приняв за ведущий элемент $a_{33}^{(2)}$, исключим x_3 из всех уравнений, кроме первых трех. Процесс такого преобразования системы продолжается до тех пор, пока исходная система (2.1) не примет треугольный вид (2.8).

При описании прямого хода метода Гаусса предполагалось, что на каждом шаге в качестве ведущих были приняты элементы, стоящие на главной диагонали и не равные нулю. Покажем, что это ограничение не существенно. Действительно, если, например, $a_{11} = 0$, то мы выберем такое i -ое уравнение, в котором коэффициент при x_1 не равен нулю ($a_{i1} \neq 0$). Такое уравнение обязательно найдется, так как в противном случае весь первый столбец матрицы системы (2.1) состоит из нулей и, следовательно, определитель этой системы равен нулю и матрица A является вырожденной. Поменяем теперь местами первое и i -ое уравнения и приняв за ведущий элемент a_{11} , проделаем все описанные выше действия и исключим x_1 из всех уравнений, кроме i -го (которое теперь является первым). Аналогично можно поступать и в случае, когда какой-либо ведущий элемент на другом шаге равен нулю.

Существуют различные видоизменения метода Гаусса, которые, сохраняя его основную идею, отличаются способом хранения матриц коэффициентов в памяти ЭВМ, способом исключения элементов, методом "борьбы" с погрешностями округления. Имеются различные модификации метода для решения задач с матрицами специального вида (симметричными, ленточными и т.д.). Доказано, что метод Гаусса – один из наиболее экономичных прямых методов для систем общего вида с точки зрения количества требуемых операций. В общем случае для решения системы n уравнений требуется $\approx n^3/3$ операций. Для понимания возможностей метода Гаусса оценим, как порядок системы n

* Случай $a_{11} = 0$ будет разобран ниже.

влияет на общее время решения системы и величину требуемой оперативной памяти. Рассмотрим пример.

Посчитаем, сколько времени понадобится для решения 10^4 уравнений с плотно заполненной матрицей на компьютере со средним быстродействием 10^6 операций в секунду. Требуемое количество арифметических операций приблизительно будет равно $N \sim \frac{1}{3} 10^{12}$. Тогда время решения задачи на компьютере примерно равно

$$t \approx \frac{1}{3} \cdot 10^{12} / 10^6 \text{ сек} = \frac{1}{3} 10^6 \text{ сек} \approx 93 \text{ часа} \approx 3,8 \text{ суток}$$

Для хранения матрицы потребуется примерно $4 \cdot n^2$ байтов, что в рассматриваемом случае составляет $4 \cdot 10^8$ байт = 400 гигабайт (предполагается, что для хранения одного числа используется 4 байта). Для сравнения укажем, что память в Winchester компьютеров Pentium 2 составляет $2 \cdot 10$ гигабайт. Видно, что решать рассматриваемую задачу на таких компьютерах нереально. Для случая линейной системы при $n = 10^3$ мы получим время счета порядка

$t \approx \frac{1}{3} \cdot \frac{10^9}{10^6} \text{ сек} = \frac{1}{3} \cdot 10^3 \text{ сек} \approx 0,1 \text{ часа} = 6 \text{ минут}$ и объем памяти $4 \cdot 10^6$ байт = 4 гигабайта. Эти оценки показывают, что на современных персональных компьютерах методом Гаусса можно решать линейные системы с плотно заполненной матрицей порядка $n \sim 1000$. С другой стороны, как отмечалось выше, во многих научных и инженерных задачах приходится решать системы порядка десятков и сотен тысяч уравнений. Это удастся делать потому, что эти матрицы в подавляющем большинстве случаев являются сильно разреженными и воспроизводимыми, и для их решения удастся использовать либо специальные модификации метода Гаусса, либо итерационные методы.

Очень существенным является вопрос об устойчивости алгоритма метода Гаусса по отношению к погрешностям округления, особенно если учесть большое количество операций, которые приходится выполнять при больших n . Показано, что погрешности округления могут сильно исказить решение, если какие-либо из ведущих элементов, на которые происходит деление во время прямого хода метода Гаусса, не равны нулю, но достаточно малы. Для уменьшения влияния погрешностей разработаны различные способы. Один из наиболее часто применяемых – метод Гаусса с частичным выбором ведущего элемента, идея которого состоит в том, что в качестве ведущего элемента в каждом рассматриваемом столбце выбирается максимальный по модулю элемент в этом столбце. Описанный выше алгоритм прямого хода метода Гаусса фактически реализуется следующим образом. На первом шаге в первом столбце отыскивается максимальный по модулю элемент. Пусть, например, это будет элемент a_{i1} . Поменяем i -ое и первое уравнения местами, так что максимальный элемент окажется в первой строке и первом столбце. Затем алгоритм исключения применяется ко всем уравнениям, кроме первого (фактически это будет i -ое уравнение). На следующем шаге мы должны исключить x_2 во всех уравнениях, кроме первых двух. Рассмотрим во втором столбце все элементы, начиная со второго, и выберем из них наибольший по модулю элемент. Пусть это будет элемент a_{k2} . Поменяем местами второе и k -ое уравнение, так что элемент a_{k2} станет теперь ведущим элементом, который

расположен во второй строке и втором столбце. Исключим теперь x_2 во всех уравнениях, кроме первых двух. Аналогичный процесс будем повторять до тех пор, пока исходная матрица будет приведена к треугольному виду. Показано, что применение такого алгоритма обеспечивает слабое влияние погрешностей округления на окончательный результат, если линейная система является хорошо обусловленной (см. 2.4). Рассмотрим простой пример.

Пример 4.

Решить систему уравнений

$$\begin{cases} 0,0001u + v = 1 \\ u + v = 2 \end{cases} \quad (2.14)$$

В обычном методе Гаусса при исключении неизвестной u из второго уравнения ведущий элемент равен 0,0001. Вычитая из второго уравнения первое, умноженное на 10000, приведем второе уравнение к виду (без учета округления)

$$- 9999 \cdot v = - 9998, \quad (2.15)$$

откуда $v = 0,99990$. Если мы теперь будем проводить округление до трех значащих цифр, то вместо (2.15) из (2.14) получим $- 10000 \cdot v = - 10000$, откуда $v = 1$. Разница решения для v получилась небольшой. С точностью до трех значащих цифр значения $v = 1$ и $v = 0,99990$ совпадают. Однако посмотрим, что произойдет, если мы теперь из первого уравнения (2.14) определим значение неизвестной величины u . Беря истинное значение $v = 0,9999$ (без учета округления), получим $0,0001 \cdot u + 0,9999 = 1$, откуда $u = 1$. С другой стороны, если взять полученное с учетом округления до трех значащих цифр значение $v = 1$, то получим $0,0001 \cdot u + 1 = 1$ и, следовательно, $u = 0$. Подстановка полученного с округлением решения $u = 0, v = 1$ в систему (2.14) показывает, что оно не удовлетворяет системе и является ошибочным.

В данном случае система (2.14) является хорошо обусловленной, и причиной получения ошибочного решения является исключение переменной u с помощью очень маленького ведущего элемента. Легко проверить, что если поменять местами первое и второе уравнения в системе (2.14), так что ведущий элемент станет равным единице, то гауссовский метод исключения приведет с учетом округления к правильному решению $u = 1, v = 1$.

Погрешности округления, совершенные в процессе вычислений по методу Гаусса, приводят к тому, что вычисленное решение отличается от точного теоретического решения $\bar{x}_* = A^{-1}\bar{b}$. Как оценить отклонение точного решения $\bar{x}_*$ от приближенного полученного решения $\bar{x}$? Обычно рассматривают две меры такого отклонения: а) точное значение вектора ошибки

$$\bar{\varepsilon} = \bar{x} - \bar{x}_* \quad (2.16)$$

и б) величину вектора невязки уравнения (2.2), равного разности между правой и левой частями уравнения при подстановке в них решения

$$\bar{r} = A\bar{x} - \bar{b} \quad (2.17)$$

Ясно, что точное значение ошибки (2.16) получить нельзя, так как нам неизвестно точное значение решения $\bar{x}_*$. В то же время невязка уравнения $\bar{r}$ зависит только от полученного приближенного решения $\bar{x}$ и может быть вычислена. Из теории линейной алгебры следует, что если одна из этих величин равна нулю, то равна нулю и другая величина. Однако из малости одной из них в общем случае не следует малость другой. Как правило, из того что $\bar{\varepsilon} \approx 0$ следует, что $\bar{r} \approx 0$, однако обратное утверждение справедливо только для хорошо обусловленных систем. Для плохо обусловленных систем при малой невязке $\bar{r} \approx 0$ погрешность решения $\bar{\varepsilon}$ может быть большой.

2.3. Другие прямые методы

Некоторой модификацией метода Гаусса является прямой метод, основанный на представлении исходной матрицы A в виде произведения двух треугольных матриц.

$$A = LU, \quad (2.18)$$

где L – нижняя треугольная матрица, а U – верхняя треугольная матрица.

Матрицы L и U имеют следующий вид:

$$L = \begin{pmatrix} 1 & & & & 0 \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ & & \ddots & \ddots & \\ l_{n1} & l_{n2} & l_{n3} & & 1 \end{pmatrix} \quad U = \begin{pmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ & u_{22} & u_{23} & \cdots & u_{2n} \\ & & u_{33} & \cdots & u_{3n} \\ & & & \ddots & \cdots \\ 0 & & & & u_{nn} \end{pmatrix} \quad (2.19)$$

и могут быть получены непосредственно в процессе применения метода Гаусса к исходной системе (2.2) при условии, что все ведущие элементы не равны нулю (и, следовательно, матрица A является невырожденной). Матрица U – это матрица треугольной линейной системы (2.8), которая является конечным результатом применения метода Гаусса к исходной линейной системе (2.2), а коэффициенты l_{ij} матрицы L , лежащие ниже главной диагонали, являются множителями, с которыми j -ая строка матрицы A вычитается из i -ой строки в процессе сведения исходной системы к треугольному виду в методе Гаусса. Разложение (2.18) называют LU -разложением матрицы A .

Для примера приведем LU – разложение матрицы для линейной системы (2.9) примера 3 предыдущего параграфа.

Пример 5.

LU -разложение матрицы A имеет вид

$$\begin{pmatrix} 2 & 1 & -1 \\ 1 & -1 & 2 \\ 3 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 3/2 & -1/3 & 1 \end{pmatrix} \cdot \begin{pmatrix} 2 & 1 & -1 \\ 0 & -3/2 & 5/2 \\ 0 & 0 & 10/3 \end{pmatrix}$$

матрица A

матрица L

матрица U

Поясним вычисление элементов матрицы L . $l_{21} = 1/2$ и $l_{31} = 3/2$, так как в процессе исключения x_1 из двух последних уравнений системы (2.9) из второго вычиталось первое, умноженное на $1/2$, а из третьего – первое, умноженное на $3/2$. При исключении x_2 из третьего уравнения (2.10) из третьего уравнения вычиталось второе, умноженное на $-1/2$, и поэтому $l_{32} = -1/3$. Непосредственным умножением матрицы L на матрицу U можно убедиться, что в результате получится матрица A .

Применение LU -разложения особенно удобно в тех случаях, когда нам необходимо многократно решать линейные системы вида

$$Ax = \bar{b} \quad (2.20)$$

с одной и той же матрицей A и различными правыми частями $\bar{b}$. В этом случае, найдя один раз LU -разложение матрицы A , мы можем сравнительно просто находить решение для любой правой части.

Действительно, найдя матрицы L и U , мы запишем систему (2.20) в виде

$$LU\bar{x} = \bar{b} \quad (2.21)$$

Обозначим вектор $U\bar{x}$ через $\bar{y}$. Тогда

$$\bar{y} = U\bar{x} \quad (2.22)$$

$$L\bar{y} = \bar{b} \quad (2.23)$$

Видно, что решение системы (2.21) при заданном $\bar{b}$ сводится к двум последовательным шагам: определению промежуточного решения $\bar{y}$ из системы (2.23) и нахождению окончательного решения $\bar{x}$ из системы (2.22). Так как матрицы L и U треугольные, то решение систем (2.22) и (2.23) не представляет труда и требует примерно n^2 операций для систем порядка n . Для хранения матриц L и U не требуется дополнительной памяти ЭВМ, так как их можно расположить на месте исходной матрицы A , если учесть, что в матрице L на главной диагонали стоят единицы, которые не нужно хранить в памяти.

Перечислим некоторые другие прямые методы, применяемые на практике.

Метод квадратного корня применяется для решения системы линейных уравнений с симметричной матрицей A . Матрица A представляется в виде произведения трех матриц

$$A = C^T D C,$$

где C - верхняя треугольная матрица,

C^T - транспонированная к ней,

D - диагональная матрица с элементами d_{ii} , равными +1 или -1.

Элементы матриц C и D вычисляются с помощью простых соотношений.

Метод Жордана отличается от метода Гаусса тем, что исходная матрица сводится не к верхней треугольной матрице, а к более простой диагональной матрице. Это требует больше операций при преобразовании системы, но зато позволяет быстрее находить решение преобразованной системы.

Клеточные методы используются для решения больших линейных систем, когда возникают трудности с размещением элементов матрицы в оперативной памяти компьютера. Они основаны на том, что решение линейной системы высокого порядка можно свести к последовательному решению нескольких линейных систем меньшего порядка.

Подробное описание этих и многих других прямых методов можно найти в [3], [10].

2.4. Обусловленность систем линейных уравнений

Рассмотрим задачу решения систем линейных уравнений

$$Ax = \bar{b}. \quad (2.24)$$

На практике обычно правые части системы b_i ($i = 1, 2, \dots, n$) и элементы матрицы a_{ij} бывают известны лишь приближенно (например, они получены с помощью измерений с некоторой погрешностью). Возникает вопрос: как неточно заданные правые части и коэффициенты системы влияют на получаемое решение? Другими словами, нас интересует, насколько изменится решение системы (2.24), если мы незначительно изменили правые части (вектор $\bar{b}$), или коэффициенты матрицы A , или и то, и другое одновременно. Для задачи (2.24) это, в сущности, вопрос о том, когда задача (2.24) является "хорошей" с точки зрения вычислений, а когда "плохой". Вопрос о "хороших" и "плохих" задачах, "хороших" и "плохих" алгоритмах уже обсуждался в главе 1 (раздел 1.4). Мы не будем проводить здесь подробное обсуждение этого вопроса, требующее введения новых понятий и доказательства ряда теорем, а приведем окончательные результаты, позволяющие ответить на заданный вопрос.

Оценки показывают, что существуют следующие соотношения между величиной относительной ошибки решения ε_x и величинами относительной погрешности задания правой части ε_b и коэффициентов системы ε_A .*

*) Здесь не будет даваться точного описания того, как определяются числа ε_x , ε_b и ε_A .

$$\varepsilon_x \leq C(A) \cdot \varepsilon_b, \quad \varepsilon_x \leq C(A) \cdot \varepsilon_A, \quad (2.25)$$

где $C(A)$ – число, зависящее от вида матрицы A , называемое степенью обусловленности матрицы A . По величине $C(A)$ можно судить о том, насколько матрица “хороша” с точки зрения вычислений. Отметим, что влияние погрешности правой части и погрешности вычисления элементов матрицы A на решение определяется одной и той же величиной – числом $C(A)$.

Можно ввести следующую условную классификацию систем линейных уравнений в зависимости от степени обусловленности матрицы ее коэффициентов:

$C = 1+10$ – хорошо обусловленные системы;

$C = 10+10^3$ – удовлетворительно обусловленные системы;

$C = 10^3+10^5$ – плохо обусловленные системы;

$C > 10^5$ – почти вырожденные системы.

Для хорошо обусловленных систем малые ошибки в задании правых частей и коэффициентов системы приводят к малым ошибкам в решении.

Плохо обусловленные системы – системы, у которых малое изменение правой части системы или коэффициентов при неизвестных (элементов матрицы) может привести к значительному изменению решения системы. Решать такие системы любыми методами весьма затруднительно. Получение плохо обусловленной системы иногда, по-видимому, указывает на то, что физическая задача поставлена недостаточно корректно.

Непосредственное определение степени обусловленности для конкретной матрицы в общем случае является достаточно сложной задачей и здесь рассматриваться не будет. Сравнительно просто величина C определяется для симметричных матриц:

$$C = |\lambda_1|/|\lambda_n|, \quad (2.26)$$

где λ_1 и λ_n – наибольшее и наименьшее по абсолютной величине собственные значения симметричной матрицы A . Приведем простой пример, иллюстрирующий понятие обусловленности.

Пример 6.

Рассмотрим систему линейных уравнений

$$\begin{cases} x_1 + 0,99x_2 = 1,99 \\ 0,99x_1 + 0,98x_2 = 1,97 \end{cases} \quad (2.27)$$

Она имеет точное решение $x_1 = 1$, $x_2 = 1$. В этом случае величину C можно непосредственно вычислить. Для определения собственных значений получим квадратное уравнение

$$\begin{vmatrix} 1-\lambda & 0,99 \\ 0,99 & 0,98-\lambda \end{vmatrix} = (1-\lambda) \cdot (0,98-\lambda) - 0,99^2 = 0,$$

откуда приближенно получим

$$\lambda_1 = 1,98005, \quad \lambda_2 = -0,00005, \quad C = |\lambda_1|/|\lambda_2| \approx 39600.$$

Таким образом, система (2.27) является плохо обусловленной. Действительно, можно рассмотреть систему

$$\begin{cases} x_1 + 0,99x_2 = 1,989903 \\ 0,99x_1 + 0,98x_2 = 1,970106 \end{cases}, \quad (2.28)$$

матрица которой совпадает с матрицей системы (2.27), а правые части отличаются весьма незначительно: $\delta b_1 = -0,000097$; $\delta b_2 = 0,000106$. Однако система (2.28) имеет решение $\{x_1 = 3,0000; x_2 = -1,0203\}$, которое совершенно не совпадает с решением системы (2.27) $\{x_1 = 1; x_2 = 1\}$. В данном примере легко усмотреть смысл сильного изменения решения при незначительном изменении правых частей: решением системы является точка пересечения двух почти параллельных прямых.

В заключение заметим, что нельзя судить о плохой обусловленности системы по малости величины ее определителя. Например, матрица n -го порядка

$$A = \begin{bmatrix} 0,1 & 0 & 0 & \dots & 0 \\ 0 & 0,1 & 0 & \dots & 0 \\ 0 & 0 & 0,1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0,1 \end{bmatrix},$$

у которой на главной диагонали все элементы равны 0,1, а остальные элементы равны нулю, очень хорошо обусловлена. Все ее собственные значения равны $\lambda_1 = \lambda_2 = \dots = \lambda_n = 0,1$ и, следовательно, $C = |\lambda_1|/|\lambda_n| = 1$. В то же время ее определитель равен $\det(A) = 10^{-n}$ и при больших n определитель очень мал. Эта почти вырожденная матрица оказывается хорошо обусловленной. Определитель – очень плохой показатель обусловленности матрицы в силу того, что он зависит от порядка матрицы (как в рассмотренном выше примере) и от масштабирования решаемой линейной системы. Под масштабированием системы понимается умножение обеих частей каждого уравнения системы на какие-либо числа. Решение системы при этом, конечно, не меняется, хотя определитель матрицы системы может принимать любое значение в зависимости от масштабных множителей.

На первый взгляд может показаться, что для проведения итераций требуется много вспомогательной работы, которую надо проводить каждый раз заново. Однако это не так. Имеется много хорошо разработанных приемов для выполнения всех перечисленных действий. Некоторые из них будут рассмотрены ниже.

3.1.2. Выбор начального приближения

Первый из этих вопросов для систем линейных уравнений оказывается не очень существенным, так как, если итерационный метод сходится, то он сходится при любом начальном приближении.* Поэтому обычно в качестве начального приближения берется какой-либо известный вектор, например, вектор правых частей $\bar{b}$ или нулевой вектор. В первом случае полагают

$$x_1^{(0)} = b_1, x_2^{(0)} = b_2, \dots, x_n^{(0)} = b_n, \quad (3.2)$$

а во втором

$$x_1^{(0)} = 0, x_2^{(0)} = 0, \dots, x_n^{(0)} = 0. \quad (3.3)$$

Конечно, если из существа решаемой задачи известно более точное приближение к решению, то надо воспользоваться им, так как в этом случае для получения решения с заданной точностью потребуется меньше итераций.

3.1.3. Приведение системы к виду, удобному для итераций

Сведение линейной системы к виду, удобному для итерации – специальный способ записи линейной системы, в котором каждая искомая переменная явно выражается через все остальные путем преобразования исходной системы.

Для проведения итераций надо записать исходную линейную систему (3.1) в таком виде, чтобы на каждой итерации можно было определять уточненное значение для каждой из неизвестных $x_1, x_2, \dots, x_n$. Для этого надо из n уравнений (3.1) получить n явных выражений для каждой из переменных через остальные переменные

[illegible]

Здесь $\varphi_1, \varphi_2, \dots, \varphi_n$ – произвольные линейные функции переменных $x_1, x_2, \dots, x_n$, причем некоторые переменные могут отсутствовать в правых

* Этим линейные системы отличаются от нелинейных, у которых сходимость существенно зависит от выбора начального приближения.

частях (3.4), но каждая переменная обязательно должна присутствовать в левой части одного уравнения. Понятно, что система (3.1) может быть записана в виде (3.4) различными способами. Рассмотрим простой пример системы второго порядка.

Пример 7.

Записать в виде, удобном для итераций, систему уравнений

$$\begin{cases} 10x_1 - x_2 = 9 \\ x_1 + 5x_2 = 6 \end{cases} \quad (3.5)$$

Один из способов заключается в том, чтобы из первого уравнения выразить x_1 , а из второго – x_2 . Получим

$$\begin{cases} x_1 = 0,1 \cdot (9 + x_2) \\ x_2 = 0,2 \cdot (6 - x_1) \end{cases} \quad (3.6)$$

Выразив, наоборот, из первого уравнения (3.5) переменную x_2 , а из второго уравнения x_1 , получим другой вид системы, подготовленной для итераций

$$\begin{cases} x_2 = 10x_1 - 9 \\ x_1 = 6 - 5x_2 \end{cases} \quad (3.7)$$

Перенеся выражения из правых частей уравнений в левую и прибавляя к обеим частям первого уравнения x_1 , а к обеим частям второго уравнения x_2 , получим еще один возможный способ записи

$$\begin{cases} x_1 = 11x_1 - x_2 - 9 \\ x_2 = 6x_2 + x_1 - 6 \end{cases} \quad (3.8)$$

Следует подчеркнуть следующее важное обстоятельство. Хотя различные виды записи системы (3.5), приведенные выше, эквивалентны в том смысле, что они имеют одно и то же решение, свойства систем (3.6) – (3.8) с точки зрения сходимости итерационного процесса совершенно различны. Позже мы увидим, что метод итераций, основанный на (3.6), быстро сходится к точному решению, а метод, основанный на (3.7), наоборот, расходится, и с его помощью получить решение нельзя. В 3.4 будут приведены достаточные условия сходимости некоторых методов, откуда будет ясно, как надо записывать линейную систему (3.1), чтобы итерационный метод сходил.

Рассмотрим стандартный прием приведения общей системы (3.1) к виду (3.4). Для этого выразим из первого уравнения x_1 через остальные переменные, из второго уравнения – x_2 , из третьего – x_3 и т.д.

(3.9)

Далее будут изложены итерационные методы, основанные на записи их в виде (3.9). Для более компактной записи представим общий вид i -го уравнения (3.9)

(3.10)

Из (3.10) видно, что в правую часть уравнения не входит переменная x_i . Рассмотрим теперь, как организовать итерационный процесс на основании формул (3.10)

Сначала рассмотрим этот метод на простом примере.

Пример 8.

Решить систему (3.5), записав ее для проведения итераций в виде (3.6)

(3.11)

Очевидно, что система (3.5) (или эквивалентная ей система (3.6)) имеет решение $x_1 = 1, x_2 = 1$. Зная точное решение, мы можем на этом примере проследить, как в процессе итераций будут изменяться приближенные решения. Зададимся начальным приближением $x_1 = 0, x_2 = 0$. Подставим его в правую часть (3.11) и определим следующее приближение (первое) по формулам

37

Теперь в качестве исходного приближения примем $x_1^{(1)} = 0,9$ и $x_2^{(1)} = 1,2$ и, подставив их в правую часть (3.11), определим следующее приближение

$$\begin{cases} x_1^{(2)} = 0,9(9 + x_2^{(1)}) = 1,02 \\ x_2^{(2)} = 0,2(6 - x_1^{(1)}) = 1,02 \end{cases}$$

Вообще, имея $(k-1)$ -ое приближение $\{x_1^{(k-1)}, x_2^{(k-1)}\}$, уточняем его с помощью формул

$$\begin{cases} x_1^{(k)} = 0,9(9 + x_2^{(k-1)}) \\ x_2^{(k)} = 0,2(6 - x_1^{(k-1)}) \end{cases} \quad (3.12)$$

Формулы (3.12) образуют одну итерацию. Многократное повторение итераций (3.12) образует для данной системы (3.11) метод простой итерации. Приведем результаты, получаемые с помощью итерационного метода (3.12), в виде таблицы 3.1.

Таблица 3.1

k	0	1	2	3	4
$x_1^{(k)}$	0	0,9	1,02	1,002	0,9996
$x_2^{(k)}$	0	1,2	1,02	0,996	0,9996
$ x_1 - x_1^{(k)} $	1	0,1	0,02	0,002	0,0004
$ x_2 - x_2^{(k)} $	1	0,2	0,02	0,004	0,0004
$ x_1^{(k)} - x_1^{(k-1)} $		0,9	0,12	0,018	0,006
$ x_2^{(k)} - x_2^{(k-1)} $		1,2	0,18	0,024	0,0036
$ r_1^{(k)} $		1,2	0,18	0,024	0,0036
$ r_2^{(k)} $		0,9	0,12	0,018	0,006

В каждом столбце таблицы указывается номер итерации, значения приближенного решения для k -ой итерации $x_1^{(k)}$ и $x_2^{(k)}$ и их отличие от точного решения $\varepsilon_{x_1} = |x_1 - x_1^{(k)}|$, $\varepsilon_{x_2} = |x_2 - x_2^{(k)}|$, где x_1, x_2 – точное решение (для системы (3.5) $x_1 = 1, x_2 = 1$). Видно, что в данном случае приближенные решения сходятся к точному, причем погрешность быстро убывает.

Рассмотрим вопрос о том, когда надо заканчивать итерационный процесс. Судить о малости погрешности в общем случае мы не можем, так как в отличие от данного модельного примера, где погрешность была приведена для иллюстрации, точного решения мы не знаем. Одним из возможных условий

где ε_1 – задаваемое малое число, характеризующее точность расчета.

$$\begin{cases} 10x_1^{(k)} - x_2^{(k)} - 9 = r_1^{(k)} \\ x_1^{(k)} + 5x_2^{(k)} - 6 = r_2^{(k)} \end{cases} \quad (3.14)$$
$$\left| r_1^{(k)} \right| < \varepsilon, \quad \left| r_2^{(k)} \right| < \varepsilon, \quad (3.15)$$

В таблице 3.1 наряду с погрешностями приведены величины, входящие в (3.13), (3.14) и (3.15).

$$\left\{ \begin{aligned} x_1^{(k)} &= \frac{1}{a_{11}}(b_1 - a_{12}x_2^{(k-1)} - a_{13}x_3^{(k-1)} - \dots - a_{1n}x_n^{(k-1)}) \\ x_2^{(k)} &= \frac{1}{a_{22}}(b_2 - a_{21}x_1^{(k-1)} - a_{23}x_3^{(k-1)} - \dots - a_{2n}x_n^{(k-1)}) \\ &\vdots \\ x_n^{(k)} &= \frac{1}{a_{nn}}(b_n - a_{n1}x_1^{(k-1)} - a_{n2}x_2^{(k-1)} - \dots - a_{n,n-1}x_{n-1}^{(k-1)}) \end{aligned} \right. \quad (3.16)$$
$$x_i^{(k)} = \frac{1}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k-1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k-1)}), (i = 1, 2, \dots, n). \quad (3.17)$$

В качестве условия окончания итерационного процесса можно, например, использовать условие

$$\|x_i^{(k)} - x_i^{(k-1)}\| < \varepsilon \quad (i = 1, 2, \dots, n), \quad (3.18)$$

которое означает, что разность значений двух последовательных приближений для всех переменных x_i должна быть меньше задаваемой величины ε .

Неудобство реализации метода простой итерации на ЭВМ заключается в том, что в памяти ЭВМ приходится хранить два массива чисел, каждый длиной n : значения переменных $x_1, \dots, x_n$ на $(k-1)$ -ой и k -ой итерациях. Метод, приводимый ниже, свободен от этого недостатка.

3.3. Метод Зейделя

Метод Зейделя – итерационный метод, при котором уже полученные на данной итерации уточненные значения искомых переменных, используются при вычислении остальных искомых переменных на этой итерации.

Метод Зейделя является естественным обобщением метода простой итерации. Анализируя формулы (3.16) (или (3.17)), легко обратить внимание на то, что, хотя мы вычисляем уточненные значения $x_1^{(k)}, x_2^{(k)}$ и т.д., тем не менее, при вычислении очередной $x_i^{(k)}$ в правой части мы берем старые значения предшествующих переменных $x_1^{(k-1)}, x_2^{(k-1)}, \dots, x_{i-1}^{(k-1)}$ (эти переменные входят в первую сумму в правой части (3.17)). Метод Зейделя заключается в том, что на каждой итерации уже вычисленные значения переменных $x_1^{(k)}, x_2^{(k)}, \dots, x_{i-1}^{(k)}$ используются в правой части при вычислении $x_i^{(k)}$. Остальные переменные в правой части берутся, конечно, с предыдущей итерации $x_{i+1}^{(k-1)}, x_{i+2}^{(k-1)}, \dots, x_n^{(k-1)}$. Формулы (3.16) для метода Зейделя запишутся в виде

$$\left\{ \begin{aligned} x_1^{(k)} &= \frac{1}{a_{11}} (b_1 - a_{12}x_2^{(k-1)} - a_{13}x_3^{(k-1)} - \dots - a_{1n}x_n^{(k-1)}) \\ x_2^{(k)} &= \frac{1}{a_{22}} (b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k-1)} - \dots - a_{2n}x_n^{(k-1)}) \\ &\dots\dots\dots \\ x_n^{(k)} &= \frac{1}{a_{nn}} (b_n - a_{n1}x_1^{(k)} - a_{n2}x_2^{(k)} - \dots - a_{n,n-1}x_{n-1}^{(k)}) \end{aligned} \right. \quad (3.19)$$

или в компактном виде для произвольной переменной

$$x_i^{(k)} = \frac{1}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^n a_{ij}x_j^{(k-1)}), (i = 1, 2, \dots, n). \quad (3.20)$$

Решим методом Зейделя уже рассмотренную выше систему второго порядка (3.5), которая в примере 8 была решена методом простой итерации.

Пример 9.

Решить методом Зейделя систему (3.5) ,

$$\begin{cases} 10x_1 - x_2 = 9 \\ x_1 + 5x_2 = 6 \end{cases},$$

приведя ее к виду, удобному для итераций, способом (3.6)

$$\begin{cases} x_1 = 0,1 \cdot (9 + x_2) \\ x_2 = 0,2 \cdot (6 - x_1), \end{cases}$$

Принимая в качестве начального приближения $x_1^{(0)} = 0$, $x_2^{(0)} = 0$, проведем несколько итераций, результаты которых сведены в таблицу. Обозначения в таблице 3.2 такие же, как и в таблице 3.1.

Таблица 3.2

k	0	1	2	3	4
$x_1^{(k)}$	0	0,9	1,002	1,0002	1,00002
$x_2^{(k)}$	0	1,02	0,9996	0,99996	0,999996
$ x_1 - x_1^{(k)} $	1	0,1	0,002	0,0002	0,00002
$ x_2 - x_2^{(k)} $	1	0,02	0,0004	0,00004	0,000004

Сравнение метода Зейделя (таблица 3.2) с методом простой итерации (таблица 3.1) показывает, что в данном случае метод Зейделя сходится быстрее (через 4 итерации погрешность решения, полученного методом Зейделя, будет для x_1 меньше в двадцать раз, а для x_2 – в сто раз). Во многих задачах дело обстоит таким же образом, хотя разница в скорости сходимости не всегда может быть настолько большой. Более быструю сходимость метода Зейделя легко понять, если учесть, что уже вычисленные более точные значения переменных сразу используются при вычислении следующих переменных на этом же итерационном шаге.

Метод Зейделя обладает еще одним преимуществом при решении задачи на ЭВМ. В отличие от метода простой итерации, здесь надо хранить в памяти ЭВМ не два массива длиной n , а один. До начала очередной итерации в этом массиве находится исходное приближенное решение $\{x_1^{(k-1)}, x_2^{(k-1)}, \dots, x_n^{(k-1)}\}$. Затем по мере вычисления уточненных значений переменных $x_1^{(k)}, x_2^{(k)}, \dots$ они засылаются в тот же массив на место старых значений соответствующих переменных. Например, после вычисления $x_1^{(k)}$ этот массив имеет вид $\{x_1^{(k)}, x_2^{(k)}, \dots, x_i^{(k)}, x_{i+1}^{(k-1)}, x_{i+2}^{(k-1)}, \dots, x_n^{(k-1)}\}$. Выражение (3.20) показывает, что именно эти переменные нужны в методе Зейделя для вычисления следующей переменной $x_{i+1}^{(k)}$ на данном итерационном шаге.

3.4. Достаточные условия сходимости методов простой итерации и Зейделя

Начнем с простого примера расходимости методов.

Пример 10.

Рассмотрим опять систему второго порядка (3.5), подготовив ее для итераций в виде (3.7)

$$\begin{cases} x_2 = 10x_1 - 9 \\ x_1 = 6 - 5x_2 \end{cases}$$

Принимая за начальное приближение $x_1^{(0)} = 0$, $x_2^{(0)} = 0$, проведем несколько итераций методом Зейделя и представим результаты в виде таблицы.

Таблица 3.3

k	0	1	2	3
$x_1^{(k)}$	0	6	-219	11001
$x_2^{(k)}$	0	51	-2199	110001

Видно, что в данном случае метод расходится, и получить решение с его помощью невозможно.

Рассмотрим теперь общий случай сходимости методов простой итерации и Зейделя. Аккуратное математическое исследование сходимости обоих методов матричными методами показывает, что условия сходимости в общем случае у них различны. Возможны случаи, когда один из них сходится, а второй – нет. Однако существует достаточное условие сходимости, справедливое для обоих методов в том случае, когда система уравнений (3.1) записана для проведения итераций в стандартном виде (3.9).

Дадим предварительно одно определение: **матрица A имеет диагональное преобладание, если в каждой строке матрицы элемент, стоящий на главной диагонали, по модулю больше или равен сумме модулей остальных элементов данной строки**

$$\begin{cases} |a_{11}| \geq |a_{12}| + |a_{13}| + \dots + |a_{1n}| \\ |a_{22}| \geq |a_{21}| + |a_{23}| + \dots + |a_{2n}| \\ \dots \dots \dots \dots \dots \dots \dots \dots \dots \\ |a_{nn}| \geq |a_{n1}| + |a_{n2}| + \dots + |a_{n,n-1}| \end{cases} \quad (3.21)$$

Например, из двух матриц

$$A = \begin{pmatrix} 5 & 2 & -1 \\ 3 & 4 & 1 \\ -8 & 6 & 15 \end{pmatrix} \quad \text{и} \quad B = \begin{pmatrix} 3 & 1 & 1 \\ -4 & 5 & 2 \\ 0 & 1 & 8 \end{pmatrix}$$

матрица A имеет диагональное преобладание, так как $|5| \geq |2| + |-1|$, $|4| \geq |3| + |1|$, $|15| \geq |-8| + |6|$, а матрица B – нет, так как во второй строке $|5| < |-4| + |2| = 6$.

Приведем теперь достаточные условия сходимости (**условия сходимости итерационного метода** – математические соотношения, при выполнении которых последовательность значений, получаемых в итерационном процессе, сходится к точному решению системы линейных уравнений). Если матрица A исходной линейной системы (3.1) имеет диагональное преобладание, причем хотя бы одно из неравенств (3.21) является строгим, то оба метода (метод простой итерации и метод Зейделя) сходятся, если итерации проводятся с помощью формул (3.17) и (3.20). При этом метод Зейделя сходится быстрее, чем метод простой итерации.

Рассмотрим пример применения этих условий.

Пример 11.

Требуется решить систему уравнений методом итераций

$$\begin{cases} 2x_1 + x_2 - 3x_3 = 0 \\ 4x_1 - x_2 - x_3 = 2 \\ 2x_1 + 5x_2 + 3x_3 = 10 \end{cases} \quad (3.22)$$

Решение. Если мы запишем (3.22) непосредственно в виде (3.9), выражая из первого уравнения x_1 , из второго – x_2 , а из третьего – x_3

$$\begin{cases} x_1 = 1/2 \cdot (0 - x_2 + 3x_3) \\ x_2 = -2 + 4x_1 - x_3 \\ x_3 = 1/3 \cdot (10 - 2x_1 - 5x_2) \end{cases} \quad (3.23)$$

то метод итераций может расходиться, так как у исходной матрицы системы (3.22)

$$\begin{pmatrix} 2 & 1 & -3 \\ 4 & -1 & -1 \\ 2 & 5 & 3 \end{pmatrix}$$

не выполняются условия диагонального преобладания ($|2| < |1| + |-3| = 4$,

$|-1| < |4| + |-1| = 5$, $|3| < |2| + |5| = 7$). Однако можно заметить, что в каждом уравнении (3.22) есть такая переменная x_i , коэффициент при которой не меньше, чем сумма модулей коэффициентов при других неизвестных в этом уравнении (в первом это переменная x_3 , во втором – x_1 , в третьем – x_2). Если теперь мы просто поменяем местами уравнения (3.22), сделав второе уравнение первым, третье уравнение – вторым, а первое – третьим

$$\begin{cases} 4x_1 - x_2 - x_3 = 2 \\ 2x_1 + 5x_2 + 3x_3 = 10 \\ 2x_1 + x_2 - 3x_3 = 0 \end{cases} \quad (3.22a)$$

то матрица этой системы

$$\begin{pmatrix} 4 & -1 & -1 \\ 2 & 5 & 3 \\ 2 & 1 & 3 \end{pmatrix}$$

будет иметь диагональное преобладание, так как $|4| > |-1| + |-1| = 2$, $|5| = |2| + |3| = 5$, $|-3| = |2| + |1| = 3$ (в первом уравнении неравенство строгое, а во втором и третьем уравнениях элементы матрицы, стоящие на главной диагонали, равны сумме модулей остальных элементов в каждой из строк). Записав теперь систему (3.22a) в виде, удобном для проведения итераций (3.9), т.е. выражая из каждого i -го уравнения искомую переменную x_i , получим сходящийся итерационный процесс.

$$\begin{cases} x_1 = 1/4 \cdot (2 + x_2 + x_3) \\ x_2 = 1/5 \cdot (10 - 2x_1 - 3x_3) \\ x_3 = 1/3 \cdot (0 + 2x_1 + x_2) \end{cases} \quad (3.23a)$$

Фактически это означает, что в исходной системе (3.22) надо было из первого уравнения выразить x_3 через остальные переменные, из второго уравнения – x_1 , а из третьего – x_2 . Численные результаты, получаемые методом Зейделя, приведены в таблице 3.4.

Таблица 3.4

k	0	1	2	3	4	5
$x_1^{(k)}$	0	0,5	1,183333	1,019444	0,978426	1,001651
$x_2^{(k)}$	0	1,8	0,966667	0,925556	1,015741	1,004821
$x_3^{(k)}$	0	0,933333	1,111111	0,988148	0,990864	1,002708

Возвращаясь к рассмотренным ранее примерам (8 и 10), легко видеть, что в примере 8 условия сходимости выполнены, а в примере 10 не выполнены и, как мы видели, итерационный процесс расходится.

Без доказательства приведем следующий факт: чем больше элементы матрицы, стоящие на главной диагонали, превосходят сумму остальных элементов строки, тем быстрее сходится итерационный процесс. В ранее рассмотренном примере 9, в котором метод Зейделя применялся для решения системы (3.5)

$$\begin{cases} 10x_1 - x_2 = 9 \\ x_1 + 5x_2 = 6 \end{cases},$$

итерационный метод сходиллся очень быстро (см. таблицу 3.2). Это объясняется тем, что в каждой строке системы (3.5) на главной диагонали стоят элементы, значительно превосходящие остальные элементы ($10 \gg 1$, $5 \gg 1$).

Уравнения (3.9), записанные в виде, удобном для итераций, можно представить в компактном матричном виде

$$\bar{x} = C\bar{x} + \bar{d}. \quad (3.24)$$

Построенный на основании (3.24) итерационный метод

$$\bar{x}^{(k)} = C\bar{x}^{(k-1)} + \bar{d} \quad (3.25)$$

есть метод простой итерации. Матричная запись метода Зейделя имеет несколько более сложный вид, и мы ее здесь приводить не будем. Если (3.24) получено из исходной системы

$$A\bar{x} = \bar{b},$$

то матрица C и вектор $\bar{d}$ имеют следующий вид

$$C = \begin{bmatrix} 0 & -\frac{a_{12}}{a_{11}} & -\frac{a_{13}}{a_{11}} & \dots & -\frac{a_{1n}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & -\frac{a_{23}}{a_{22}} & \dots & -\frac{a_{2n}}{a_{22}} \\ \dots & \dots & \dots & \dots & \dots \\ -\frac{a_{n1}}{a_{nn}} & -\frac{a_{n2}}{a_{nn}} & -\frac{a_{n3}}{a_{nn}} & \dots & 0 \end{bmatrix} \quad \bar{d} = \begin{bmatrix} \frac{b_1}{a_{11}} \\ \frac{b_2}{a_{22}} \\ \dots \\ \frac{b_n}{a_{nn}} \end{bmatrix} \quad (3.26)$$

Видно, что условия диагонального преобладания для матрицы A приводят к следующим соотношениям для матрицы C

$$\begin{cases} \sum_{j=1}^n |C_{1j}| \leq 1 \\ \sum_{j=1}^n |C_{2j}| \leq 1 \\ \dots\dots\dots \dots\dots\dots \\ \sum_{j=1}^n |C_{nj}| \leq 1 \end{cases} \quad (3.27)$$

(3.27) означает, что если сумма модулей элементов в каждой строке будет не больше единицы, то итерационный метод (3.24) будет сходиться.

Завершая рассмотрение итерационных методов, отметим очень важное обстоятельство. Итерационные методы применяются, как правило, для тех случаев, когда матрица линейной системы является разреженной и порядок системы n достаточно велик ($n > 10^3 + 10^4$). Сходимость итерационных методов обычно уменьшается с ростом n . Это означает, что при больших n для получения решения с заданной точностью требуется очень большое количество итераций, в результате чего время решения для многих задач становится чрезмерно большим. Были проведены многочисленные исследования возможных способов ускорения сходимости итерационных методов для большинства встречающихся на практике задач. Под термином **ускорение сходимости итерационных методов** понимаются специальные теоретические разработки, приемы и алгоритмы, позволяющие вместо исходного итерационного метода построить другой, который сходится к искомому решению гораздо быстрее, чем исходный. Обзор и анализ многих из этих методов приведен, например, в [3], [10]. В качестве одного из примеров эффективного ускорения сходимости для многих задач, связанных с решением уравнений в частных производных, приведем широко используемый на практике метод релаксации. **Метод релаксации** – это способ ускорения сходимости итерационного процесса для решения систем линейных уравнений путем введения вспомогательной величины – коэффициента релаксации ϖ , который используется для корректировки результатов при переходе от одного итерационного шага к другому. Теоретические основы подобных методов являются достаточно сложными и требуют исследования собственных значений для матриц, определяющих переход от одного итерационного приближения к другому. Пояснить суть одной из простых реализаций метода релаксации можно следующим образом. Пусть имеется итерационный метод решения систем линейных уравнений (например, метод Зейделя), который образует последовательность $\bar{x}_0, \bar{x}_1, \bar{x}_2, \dots, \bar{x}_k, \dots$, которая сходится к точному решению. Если эта последовательность сходится медленно, то ускорить сходимость можно с помощью следующего приема. Построив по $\bar{x}_{k-1}$ следующее приближение $\bar{x}_k$, за окончательный результат на k -ом итерационном шаге берется их линейная комбинация

$$\bar{x} = \varpi \bar{x}_k + (1 - \varpi) \bar{x}_{k-1}. \quad (3.28)$$

Здесь ω – параметр релаксации. При $1 < \omega < 2$ метод называется методом верхней релаксации, а при $0 < \omega < 1$ – методом нижней релаксации. Для некоторых задач удастся теоретически получить оптимальное значение $\omega_{\text{опт}}$, которое обеспечивает максимальную скорость сходимости в методе верхней релаксации. В ряде случаев величину ω , близкую к $\omega_{\text{опт}}$, удастся определить с помощью численных экспериментов.

3.5. Сравнительная оценка прямых и итерационных методов

Выбор численного метода решения системы линейных уравнений зависит от вида решаемой задачи и типа имеющейся вычислительной техники.

Для заполненных матриц сравнительно небольшой размерности ($n < 1000$) одним из наиболее эффективных является, по-видимому, метод Гаусса и его различные модификации, так как он прост в реализации и наиболее универсален. Однако с ростом n начинают проявляться недостатки прямых методов, которые заключаются в следующем:

- они требуют, как правило, хранения всей матрицы, что является очень серьезным ограничением на объем требуемой оперативной памяти*;
- происходит постоянное накапливание погрешностей округления, так как на каждом этапе вычислений используются результаты предыдущих операций.

В случае, если матрица является разреженной, но ненулевые элементы располагаются в матрице упорядоченным образом (например, матрица является ленточной или ненулевые элементы имеют вид клеток, расположенных вдоль главной диагонали), то метод Гаусса также является достаточно эффективным, так как можно организовать программу реализации метода так, чтобы нулевые элементы не хранить в памяти ЭВМ.

Для разреженной матрицы произвольного вида метод Гаусса становится неудобным в силу того, что в процессе преобразований вместо некоторых нулевых элементов появляются ненулевые элементы, что приводит к значительному увеличению количества ненулевых элементов. В этом случае итерационные методы могут быть гораздо более эффективными, так как для них, как правило, не требуется хранить нулевые элементы и совершать с ними какие-либо операции. Более того, во многих случаях, встречающихся в практических задачах, можно вообще не хранить элементы матрицы в памяти ЭВМ, а вычислять их по мере необходимости. Другое достоинство итерационных методов заключается в том, что в них погрешность не накапливается в процессе вычислений, так как точность вычисления каждой итерации определяется лишь результатами предыдущей итерации и почти не зависит от результатов других итерационных шагов. Это свойство самокорректировки метода, когда допущенная на какой-либо итерации небольшая ошибка не влияет на конечный результат, особенно полезно при решении систем линейных уравнений высокого порядка.

Недостатком итерационных методов во многих случаях является их довольно медленная сходимость, особенно при больших n . Поэтому в вычислительной математике большое внимание уделяется вопросам ускорения

* Можно, конечно, хранить всю матрицу на жестком диске, но тогда требуется организовать постоянный обмен массивов информации между оперативной памятью и диском, что значительно увеличивает общее время решения задачи.

сходимости итерационных методов. С основными полученными результатами можно ознакомиться в монографиях [3], [10]. -

Во многих случаях оказывается удобным комбинировать применение прямых и итерационных методов. Одним из таких подходов является **итерационное уточнение решения** системы линейных уравнений, полученного каким-либо прямым методом. Изложим коротко суть этого метода. Пусть имеется система линейных уравнений, записанная в матричном виде

$$A\bar{x} = \bar{b} \quad (3.29)$$

и пусть $\bar{x}_1$ – приближенное решение, полученное каким-либо методом.

Подставим $\bar{x}_1$ в уравнение и вычислим невязку уравнения $\bar{r}_1 = \bar{b} - A\bar{x}_1$. Для точного решения $\bar{x}_*$ невязка равна нулю $\bar{r}_* = \bar{b} - A\bar{x}_* = 0$. Для приближенного решения невязка будет достаточно малым числом. Построим погрешность решения $\bar{e}_1 = \bar{x}_* - \bar{x}_1$ и получим уравнение для определения погрешности

$$A\bar{e}_1 = A(\bar{x}_* - \bar{x}_1) = A\bar{x}_* - A\bar{x}_1 = \bar{b} - (\bar{b} - \bar{r}_1) = \bar{r}_1.$$

Приведенные выкладки показывают, что погрешность $\bar{e}_1$ удовлетворяет исходному уравнению, в котором в правой части вместо вектора $\bar{b}$ стоит вектор невязки $\bar{r}_1$, вычисленный подстановкой приближенного решения $\bar{x}_1$ в исходное уравнение. Определим из уравнения

$$A\bar{e}_1 = \bar{r}_1 \quad (3.30)$$

величину $\bar{e}_1$. Если бы величина $\bar{e}_1$ определялась из (3.30) точно (без погрешностей округления), то мы определили бы точное решение $\bar{x}_* = \bar{x}_1 + \bar{e}_1$. Однако из-за наличия погрешностей знание $\bar{e}_1$ позволяет нам определить не точное решение, а уточненное значение решения.

$$\bar{x}_2 = \bar{x}_1 + \bar{e}_1. \quad (3.31)$$

Во многих случаях вычислительный алгоритм, основанный на соотношениях (3.30) и (3.31), позволяет существенно уточнить первоначальное приближенное решение $\bar{x}_1$ при условии, что невязка $\bar{r}_1$ вычисляется достаточно точно. Вообще говоря, процесс итерационного уточнения можно продолжать, определяя величины $\bar{x}_3, \bar{x}_4, \dots$ однако это требует все более точного вычисления невязок уравнения. Удобство использования метода итерационного уточнения заключается в том, что для определения $\bar{e}_1$ приходится решать систему линейных уравнений с той же исходной матрицей A . Поэтому, если для решения исходной системы $A\bar{x} = \bar{b}$ используется LU -разложение, то знание треугольных матриц L и U позволяет без труда решить систему (3.30) с помощью небольшого количества операций ($\approx n^2$ операций вместо примерно n^3 операций в методе Гаусса).

ГЛАВА 4. МАТРИЧНЫЕ ЗАДАЧИ И ПРОБЛЕМА СОБСТВЕННЫХ ЗНАЧЕНИЙ

4.1. Нахождение обратной матрицы

Напомним, что **матрицей, обратной** данной матрице A , называется матрица A^{-1} такая, что произведение A на A^{-1} равняется единичной матрице

$$A A^{-1} = A^{-1} A = E \quad (4.1)$$

Пример 12. Заданы две матрицы

$$A = \begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix} \quad B = \begin{pmatrix} 1/2 & -1/6 \\ 0 & 1/3 \end{pmatrix}$$

Проверить, будет ли матрица B обратной к A .

Решение. Вычислим произведение этих матриц

$$A \cdot B = \begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix} \cdot \begin{pmatrix} 1/2 & -1/6 \\ 0 & 1/3 \end{pmatrix} = \begin{pmatrix} 2 \cdot 1/2 + 1 \cdot 0 & -2/6 + 1/3 \\ 0 \cdot 1/2 + 3 \cdot 0 & 0 \cdot (-1/6) + 3 \cdot 1/3 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Следовательно, $B = A^{-1}$.

Теоретический метод нахождения обратной матрицы, изучаемый в курсе высшей математики и основанный на вычислении миноров матрицы A , при использовании ЭВМ неприменим при достаточно больших n по тем же причинам, что и правило Крамера для решения систем линейных уравнений (см. раздел 2.1). Для нахождения A^{-1} на компьютерах используется обращение матрицы методом Гаусса, основанное на решении системы n линейных уравнений с матрицей A с n различными правыми частями. Изложим этот метод. Запишем подробно матричное равенство (4.1), обозначая элементы обратной матрицы A^{-1} через b_{ij} .

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & & 1 \end{pmatrix} \quad (4.2)$$

В (4.2) все элементы a_{ij} матрицы A заданы, а элементы b_{ij} обратной матрицы A^{-1} подлежат определению. Выпишем на основании (4.2) уравнения, которым должны удовлетворять элементы b_{ij} . Вспоминая правило умножения матриц, получим для элементов первого столбца b_{i1} ($i = 1, 2, \dots, n$) систему уравнений

(4.3)

Аналогично, для элементов второго столбца обратной матрицы получим систему уравнений

(4.4)

Вообще, для элементов j -го столбца обратной матрицы, система уравнений имеет вид

(4.5)

Запишем расширенную матрицу, состоящую из n строк и $2n$ столбцов, которая состоит из расположенных рядом матрицы A и единичной матрицы E

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} & 1 & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & a_{2n} & 0 & 1 & \cdots & 0 \\ \cdots & \cdots & & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & 0 & 0 & & 1 \end{pmatrix} \quad (4.6)$$

Элементы матрицы A определяют коэффициенты при неизвестных в линейной системе (4.5), а столбцы единичной матрицы определяют n различных правых частей для каждой из n систем (4.5). Все n систем (4.5) будем решать методом Гаусса одновременно, сводя матрицу A к треугольному виду, причем все преобразования метода Гаусса будем проводить с "длинными" строками расширенной матрицы (4.6). После приведения матрицы A к треугольному виду расширенная матрица примет вид

$$\begin{pmatrix} \bar{a}_{11} & \bar{a}_{12} & \cdots & \bar{a}_{1n} & 1 & 0 & \cdots & 0 \\ 0 & \bar{a}_{22} & \cdots & \bar{a}_{2n} & \bar{c}_{21} & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \bar{a}_{nn} & \bar{c}_{n1} & \bar{c}_{n2} & \cdots & 1 \end{pmatrix} \quad (4.7)$$

Первые n столбцов в (4.7) составляют верхнюю треугольную матрицу, полученную из исходной матрицы A в ходе преобразований метода Гаусса. Следующие n столбцов – это преобразованные n столбцов правых частей систем уравнений (4.5). Используя (4.7), с помощью обратного хода метода Гаусса можно найти теперь решения n преобразованных систем уравнений (4.5) и, следовательно, найти n столбцов обратной матрицы. Поясним разобранный алгоритм на примере.

Пример 13. Найти обратную матрицу для матрицы A

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 5 \end{pmatrix}$$

Решение. Запишем расширенную матрицу

$$\begin{pmatrix} 2 & 1 & 1 & 0 \\ 1 & 5 & 0 & 1 \end{pmatrix}$$

и преобразуем ее так, чтобы матрица A стала верхней треугольной. Для этого из второй строки расширенной матрицы вычтем первую строку, умноженную на $1/2$. Получим

$$\begin{pmatrix} 2 & 1 & 1 & 0 \\ 0 & 9/2 & -1/2 & 1 \end{pmatrix}$$

Эта расширенная матрица фактически эквивалентна двум системам уравнений второго порядка для нахождения столбцов обратной матрицы:

$$\begin{cases} 2b_{11} + b_{21} = 1 \\ 9/2 b_{21} = -1/2 \end{cases} \quad \begin{cases} 2b_{12} + b_{22} = 0 \\ 9/2 b_{22} = 1 \end{cases}$$

Из второго уравнения первой системы находим $b_{21} = -1/9$. Подставив полученное значение b_{21} в первое уравнение, найдем

$$2b_{11} - 1/9 = 1 \Rightarrow b_{11} = 5/9$$

Аналогично из второй системы определим $b_{22} = 2/9$, и, подставив в первое уравнение, найдем $2b_{12} + 2/9 = 0 \Rightarrow b_{12} = -1/9$.

Следовательно, обратная матрица имеет вид

$$A^{-1} = \begin{bmatrix} 5/9 & -1/9 \\ -1/9 & 2/9 \end{bmatrix}.$$

Для проверки умножим матрицу A на полученную обратную матрицу

$$A \cdot A^{-1} = \begin{bmatrix} 2 & 1 \\ 1 & 5 \end{bmatrix} \cdot \begin{bmatrix} 5/9 & -1/9 \\ -1/9 & 2/9 \end{bmatrix} = \begin{bmatrix} 2 \cdot 5/9 + 1 \cdot (-1/9) & -2/9 + 2/9 \\ 1 \cdot 5/9 + 5 \cdot (-1/9) & 1 \cdot (-1/9) + 5 \cdot 2/9 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Рассмотренный алгоритм нахождения обратной матрицы реализован в большинстве стандартных программ, имеющихся в математическом обеспечении современных ЭВМ. Он является довольно эффективным, так как требует выполнения количества операций, которое всего в три раза превышает количество операций, необходимое для решения одной линейной системы такого же порядка методом Гаусса.

4.2. Вычисление определителя

В разделе 2.1 уже отмечалось, что непосредственное вычисление определителя матрицы уже при сравнительно небольших n ($n > 10$) требует

много времени ЭВМ, причем с ростом n это время растет чрезвычайно быстро. Вместе с тем очень легко вычисляется определитель верхней (нижней) треугольной матрицы – он просто равен произведению элементов матрицы, стоящих на главной диагонали. Это дает возможность удобного вычисления определителя матрицы путем сведения матрицы к треугольному виду с помощью прямого хода метода Гаусса. Эквивалентные преобразования, которые используются в методе Гаусса, не изменяют величину определителя. Знак определителя (но не его величина) может измениться только при перестановке его столбцов или строк. Такая необходимость возникает, например, в методе Гаусса с выбором в качестве ведущего элемента максимального элемента в столбце, либо в случае, когда элемент на главной диагонали равен нулю и нам приходится переставлять строки для получения ненулевого ведущего элемента. Количество таких перестановок легко запомнить в специальном счетчике в процессе приведения матрицы к треугольному виду. Значение определителя после приведения матрицы A к треугольному виду вычисляется по формуле

$$\det(A) = \pm a_{11} a_{22} \dots a_{nn}, \quad (4.8)$$

где диагональные элементы a_{ii} берутся из преобразованной, а не исходной матрицы. Знак зависит от того, сколько раз приходилось менять местами строки в процессе приведения матрицы к треугольному виду. Если количество перестановок было четным, то будет знак плюс, а если нечетным, то минус.

4.3. Нахождение собственных значений и собственных векторов матриц

Нахождение собственных значений и собственных векторов матриц требуется во многих физических и технических задачах при исследовании устойчивости различных процессов, при определении устойчивости и колебаний различных инженерных сооружений. В вычислительной математике исследования по устойчивости алгоритмов и анализу сходимости различных вычислительных методов также требуют решения задач на собственные значения.

Напомним связанные с рассматриваемой задачей определения. **Собственные значения** – для матрицы n -ого порядка это n действительных или комплексных чисел, определяемых как корни характеристического многочлена данной матрицы и характеризующих свойства матрицы при различных вычислительных преобразованиях. Матрица A n -го порядка имеет n собственных значений, которые мы обозначим через $\lambda_1, \lambda_2, \dots, \lambda_n$, и соответствующие им ненулевые собственные векторы $\bar{u}_1, \bar{u}_2, \dots, \bar{u}_n$. В общем случае собственные значения могут быть как действительными, так и комплексными числами. Если среди собственных значений встречаются два или несколько одинаковых, то они называются кратными. В этом случае собственные векторы, соответствующие кратным собственным значениям, объединяются в один вектор.

По определению собственные вектора и собственные значения удовлетворяют соотношениям

$$A\bar{u}_1 = \lambda_1 \bar{u}_1, \quad A\bar{u}_2 = \lambda_2 \bar{u}_2, \quad \dots \quad A\bar{u}_n = \lambda_n \bar{u}_n, \quad (4.9)$$

где $\vec{u}_i$ – ненулевые векторы. Смысл выражения (4.9) заключается в том, что собственные векторы – это такие ненулевые векторы, которые матрица A преобразует в коллинеарные им векторы, при этом соответствующие им собственные значения являясь коэффициентами растяжения (сжатия) длины исходного собственного вектора.

Введем координаты собственного вектора $\bar{u}_i = \{u_{i1}, u_{i2}, \dots, u_{in}\}$. Тогда соотношение (4.9) сводит нахождение собственного вектора $\bar{u}_i$ к решению системы линейных уравнений для нахождения его координат

$$\left\{ \begin{array}{ccccccc} a_{11}u_{i1} & + & a_{12}u_{i2} & + \cdots + & a_{1n}u_{in} & = & \lambda_i u_{i1} \\ a_{21}u_{i1} & + & a_{22}u_{i2} & + \cdots + & a_{2n}u_{in} & = & \lambda_i u_{i2} \\ \dots\dots & & \dots\dots & & \dots\dots & & \dots\dots \\ a_{n1}u_{i1} & + & a_{n2}u_{i2} & + \cdots + & a_{nn}u_{in} & = & \lambda_i u_{in} \end{array} \right. \quad (4.10)$$

Перенеся все члены в левую часть, получим систему линейных однородных уравнений:

$$\left\{ \begin{array}{ccccccc} (a_{11} - \lambda_i)u_{i1} & + & a_{12}u_{i2} & + \cdots + & a_{1n}u_{in} & = & 0 \\ a_{21}u_{i1} & + & (a_{22} - \lambda_i)u_{i2} & + \cdots + & a_{2n}u_{in} & = & 0 \\ \dots\dots\dots & \dots & \dots\dots\dots & \dots\dots\dots & \dots\dots\dots & \dots & \dots \\ a_{n1}u_{i1} & + & a_{n2}u_{i2} & + \cdots + & (a_{nn} - \lambda_i)u_{in} & = & 0 \end{array} \right. \quad (4.11)$$

Система (4.11) имеет ненулевые решения лишь тогда, когда определитель матрицы системы (4.11) равен нулю. Учитывая, что уравнения (4.11) справедливы для любого собственного вектора $\vec{u}_i$, то в общем случае можно записать условия существования ненулевого решения для произвольного собственного вектора в виде

$$\begin{vmatrix} a_{11} - \lambda & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} - \lambda & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} - \lambda \end{vmatrix} = 0 \quad (4.12)$$

Раскрывая определитель (4.12), мы получаем характеристический многочлен

$$C_0 \lambda^n + C_1 \lambda^{n-1} + C_2 \lambda^{n-2} + \dots + C_{n-1} \lambda + C_n = 0 \quad (4.13)$$

Корни этого многочлена являются собственными значениями $\lambda_1, \lambda_2, \dots, \lambda_n$ матрицы A . Для конкретного собственного значения λ_i система уравнений (4.11) позволяет найти координаты собственного вектора $\bar{u}_i = \{u_{i1}, u_{i2}, \dots, u_{in}\}$. Этот вектор определяется с точностью до произвольного сомножителя, так как решение однородной системы (4.11) не является единственным. Обычно при нахождении собственных векторов матрицы одну из компонент вектора полагают равным некоторому числу, например $u_{in} = 1$. Остальные компоненты определяются единственным образом из системы уравнений (4.11), в которой отброшено уравнение, являющееся следствием остальных*.

Неединственность собственного вектора легко показать, используя его определение (4.9). Действительно, если $\bar{u}$ – собственный вектор, то $\gamma \cdot \bar{u}$ – также собственный вектор, так как

$$A(\gamma \cdot \bar{u}) = \gamma \cdot A\bar{u} = \gamma \cdot \lambda \bar{u} = \lambda(\gamma \cdot \bar{u}),$$

где γ – любая скалярная величина (число).

Поскольку собственный вектор определяется с точностью до сомножителя, то его можно нормировать, задавая длину любым удобным образом. В частности, деля каждую его координату на длину вектора, мы получим собственный вектор единичной длины.

Отметим некоторые свойства собственных значений, которые нам понадобятся ниже.

1. Если собственные значения матрицы действительны и различны, то соответствующие им собственные векторы ортогональны и образуют базис рассматриваемого n -мерного пространства.
2. Все собственные значения симметричной матрицы являются действительными числами.
3. Две матрицы A и C называются подобными, если существует такая невырожденная матрица P , что

$$C = P^{-1} A P \quad (4.14)$$

Собственные значения подобных матриц совпадают.

4. Легко установить связь между собственными значениями матрицы A и обратной ей матрицы A^{-1} . Пусть $\bar{u}$ и λ – собственный вектор и собственное значение матрицы A

$$A\bar{u} = \lambda \cdot \bar{u}.$$

* На самом деле в зависимости от ранга матрицы системы (4.11) количество линейно независимых уравнений может быть меньше, чем $n - 1$. Подробно эти вопросы разобраны в курсе высшей математики.

Умножив обе части равенства на $\frac{1}{\lambda} A^{-1}$, получим

$$\frac{1}{\lambda} A^{-1} A \bar{u} = \frac{1}{\lambda} A^{-1} \lambda \bar{u} \Rightarrow \frac{1}{\lambda} E \bar{u} = A^{-1} \bar{u} \Rightarrow \frac{1}{\lambda} \bar{u} = A^{-1} \bar{u}.$$

Это означает, что собственные значения для A и A^{-1} связаны соотношением

$$\bar{\lambda} = \frac{1}{\lambda}, \quad (4.15)$$

где $\bar{\lambda}$ – собственное значение обратной матрицы A^{-1} .

Нахождение всех собственных значений и собственных векторов матрицы называется **полной проблемой собственных значений**. Нахождение лишь некоторых из собственных значений и собственных векторов называется **частичной проблемой собственных значений**.

Во многих практических задачах приходится решать **обобщенную проблему собственных значений**. Она определяется следующим образом. Пусть заданы невырожденные матрицы A и B . Требуется найти ненулевой вектор $\bar{u}$ и число λ , удовлетворяющие равенству

$$A \bar{u} = \lambda B \bar{u}. \quad (4.16)$$

Тогда $\bar{u}$ и λ соответственно называются обобщенным собственным вектором и обобщенным собственным значением.

Легко показать, что обобщенную задачу можно свести к обычной задаче на собственные значения и собственные векторы. Действительно, умножив обе части векторного равенства (4.15) слева на обратную матрицу B^{-1} , получим

$$B^{-1} A \bar{u} = \lambda B^{-1} B \bar{u} \Rightarrow B^{-1} A \bar{u} = \lambda \bar{u}.$$

Видно, что обобщенная задача сводится к обычной задаче для матрицы $B^{-1}A$.

Перейдем к вопросам численного нахождения собственных значений и собственных векторов. Эта задача является одной из наиболее сложных вычислительных задач алгебры. На первый взгляд может показаться, что определение собственных значений сводится к нахождению корней характеристического многочлена (4.13), однако сложность здесь заключается в том, что для произвольной матрицы (особенно в случае большой размерности) весьма затруднительно вычислить сами коэффициенты C_i характеристического многочлена. Дополнительные трудности возникают из-за того, что среди собственных значений часто встречаются кратные. Поэтому большинство численных методов основано не на получении характеристического многочлена (4.13) для исходной матрицы, а на различных преобразованиях этой матрицы с помощью соотношений типа (4.14) и ее сведению к более простому виду, откуда собственные значения определяются гораздо проще.

На практике для симметричных матриц чаще всего используются различные варианты метода вращений. Не приводя подробное описание метода, которое является достаточно громоздким, опишем лишь его суть. К

исходной матрице A многократно применяется преобразование (4.14), в результате чего матрица сводится к гораздо более простому виду: к диагональной матрице или к трехдиагональной матрице. Для диагональной матрицы

$$\begin{bmatrix} a'_{11} & 0 & 0 \\ 0 & a'_{22} & 0 \\ 0 & 0 & a'_{nn} \end{bmatrix}$$

собственные значения могут быть вычислены сразу

$$\lambda_1 = a'_{11}, \quad \lambda_2 = a'_{22}, \quad \dots \quad \lambda_n = a'_{nn},$$

а для трехдиагональной матрицы они вычисляются с помощью простых рекуррентных формул.

Матрица P , с помощью которой производится преобразование (4.14), подбирается таким образом, чтобы на каждом шаге преобразования в преобразованной матрице $P^{-1}AP$ появлялись два нулевых элемента, симметричных относительно главной диагонали. Такой матрицей P является матрица вращения, имеющая вид

$$P = \begin{bmatrix} 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & \dots & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ \vdots & \vdots & & \cos \varphi & \dots & -\sin \varphi & \vdots \\ \vdots & \vdots & & \dots & \ddots & \dots & \vdots \\ \vdots & \vdots & & -\sin \varphi & \dots & \cos \varphi & \vdots \\ \vdots & & & & & \ddots & \vdots \\ 0 & 0 & & & \dots & & 1 \end{bmatrix}. \quad (4.17)$$

Эта матрица отличается от единичной матрицы E только изменением четырех членов: два из них стоят на главной диагонали $p_{ii} = p_{jj} = \cos \varphi$, а два других симметричны относительно главной диагонали $p_{ij} = p_{ji} = -\sin \varphi$. Геометрический смысл умножения на матрицу (4.17) заключается в том, что при таком преобразовании осуществляется плоское вращение в n -мерном пространстве в плоскости, проходящей через оси i и j на угол φ (остальные оси координат при таком вращении остаются неподвижными). Угол поворота φ выбирается таким образом, чтобы два элемента в исходной матрице, симметричные относительно главной диагонали, обратились в нуль.

Существуют различные варианты метода вращений, которые отличаются тем, к какому виду сводится исходная матрица A , и способом организации вычислений. Подробное описание методов решения полной проблемы собственных значений приведено в [2], [3], [10].

4.4. Определение наибольшего и наименьшего собственных значений итерационным методом

В практических задачах часто нужны не все собственные значения, а лишь некоторые из них. Чаще всего (например, в вопросах устойчивости) требуется определить лишь минимальное (или максимальное) по модулю собственное значение матрицы. В этом случае нецелесообразно решать полную проблему собственных значений.

Для решения частичной проблемы собственных значений, состоящей в определении одного или нескольких собственных значений и соответствующих им собственных векторов, проще всего использовать итерационные методы. Строится итерационный процесс, который сходится к одному собственному вектору с соответствующим вычислением одного собственного значения. Опишем такой процесс для нахождения максимального по модулю собственного значения и соответствующего собственного вектора.

Рассмотрим случай, когда матрица A n -го порядка имеет все действительные собственные значения $\lambda_1, \lambda_2, \dots, \lambda_n$ и соответствующие им собственные векторы $\bar{u}_1, \bar{u}_2, \dots, \bar{u}_n$. Пусть также все собственные значения расположены в порядке убывания, причем значение λ_1 по модулю строго больше, чем остальные собственные значения

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots |\lambda_n|. \quad (4.18)$$

В предыдущем разделе 4.3 было отмечено следующее свойство: если все собственные значения матрицы действительны и различны, то соответствующие им векторы образуют базис в n -мерном пространстве. Это значит, что любой n -мерный вектор можно представить в виде линейной комбинации векторов $\bar{u}_1, \bar{u}_2, \dots, \bar{u}_n$. Возьмем произвольный вектор $\bar{y}_0$. Тогда

$$\bar{y}_0 = c_1 \bar{u}_1 + c_2 \bar{u}_2 + \dots + c_n \bar{u}_n = \sum_{i=1}^n c_i \bar{u}_i. \quad (4.19)$$

Умножим теперь матрицу A на вектор $\bar{y}_0$. В результате вектор $\bar{y}_0$ преобразуется в другой вектор $\bar{y}_1$

$$\bar{y}_1 = A \bar{y}_0. \quad (4.20)$$

Запишем в развернутом виде вектор $\bar{y}_1$, используя (4.19), (4.20) и определение собственных векторов и собственных значений

$$A\bar{u}_i = \lambda \cdot \bar{u}_i \quad (i = 1, 2, \dots, n). \quad (4.21)$$

Тогда (4.20) запишется в виде

$$\bar{y}_1 = A\bar{y}_0 = A \cdot \left(\sum_{i=1}^n c_i \bar{u}_i \right) = \sum_{i=1}^n c_i (A\bar{u}_i) = \sum_{i=1}^n c_i \lambda_i \cdot \bar{u}_i. \quad (4.22)$$

Формула (4.22) показывает, что действуя матрицей A на произвольный вектор $\bar{y}_0$, мы получаем вектор, коэффициенты разложения которого по собственным векторам умножаются на соответствующее данному собственному вектору собственное значение.

Умножим теперь матрицу A на полученный вектор $\bar{y}_1$ и получим новый вектор $\bar{y}_2$

$$\bar{y}_2 = A\bar{y}_1 = A^2 \bar{y}_0 = A \cdot \left(\sum_{i=1}^n c_i \lambda_i \cdot \bar{u}_i \right) = \sum_{i=1}^n c_i \lambda_i \cdot (A\bar{u}_i) = \sum_{i=1}^n c_i \lambda_i^2 \cdot \bar{u}_i.$$

Здесь было использовано матричное обозначение $A \cdot A\bar{y} = A^2 \bar{y}$ для второй степени матрицы. Повторяя многократное умножение матрицы A на вновь получаемые векторы, получаем

$$\bar{y}_k = A\bar{y}_{k-1} = A^k \bar{y}_0 = \sum_{i=1}^n c_i \lambda_i^k \cdot \bar{u}_i. \quad (4.23)$$

Теперь воспользуемся тем, что λ максимальное по модулю собственное значение, и преобразуем (4.23)

$$\bar{y}_k = \lambda_1^k \cdot \left[c_1 \bar{u}_1 + c_2 \left(\frac{\lambda_2}{\lambda_1} \right)^k \bar{u}_2 + c_3 \left(\frac{\lambda_3}{\lambda_1} \right)^k \bar{u}_3 + \dots + c_n \left(\frac{\lambda_n}{\lambda_1} \right)^k \bar{u}_n \right].$$

Так как ввиду (4.18)

$$\left| \frac{\lambda_2}{\lambda_1} \right| < 1, \quad \left| \frac{\lambda_3}{\lambda_1} \right| < 1, \quad \dots \quad \left| \frac{\lambda_n}{\lambda_1} \right| < 1,$$

то с ростом степени k эти величины убывают и при $k \rightarrow \infty$ все $(|\lambda_i|/|\lambda_1|)^k \rightarrow 0$. Это означает, что при достаточно больших k коэффициенты, на которые умножаются векторы $\bar{u}_2, \bar{u}_3, \dots, \bar{u}_n$, будут малы и, следовательно, вектор $\bar{y}_k$ будет почти коллинеарен первому собственному вектору

$$\bar{y}_k \approx \lambda_1^k c_1 \bar{u}_1. \quad (4.24)$$

Приведенные соотношения (4.23), (4.24) являются основой степенного метода нахождения максимального собственного значения. **Степенной метод** – итерационный метод нахождения наибольшего по модулю собственного значения матрицы и соответствующего ему собственного вектора. Метод называется степенным, так как в нем вычисляются различные степени матрицы. Однако алгоритм его реализации на практике несколько отличается от того, который был описан выше. Это вызвано тем, что при достаточно большом λ длина получаемых по формуле (4.23) векторов сильно растет. Поэтому удобно реализовать степенной метод таким образом, чтобы на каждом шаге итерации вида (4.23) проводить нормировку получаемого вектора, делая его длину равной единице (здесь мы пользуемся тем, что собственный вектор определяется с точностью до произвольного сомножителя и поэтому, разделив каждую компоненту вектора на его длину, мы получим вектор единичной длины). Один из удобных способов реализации степенного метода заключается в следующем.

Задается начальный вектор $\bar{y}_0$ с координатами $\{y_{01}, y_{02}, \dots, y_{0n}\}$. Вычисление следующего приближения – вектора $\bar{y}_1$ – состоит из трех этапов.

1. Вычисляется длина вектора $\bar{y}_0$

$$d_0 = \sqrt{\sum_{i=1}^n (y_{0i})^2}. \quad (4.25)$$

2. Производится нормировка вектора $\bar{y}_0$ путем деления каждой его компоненты на величину d_0

$$\bar{w}_0 = 1/d_0 \bar{y}_0. \quad (4.26)$$

В результате вектор $\bar{w}_0$ будет иметь длину, равную единице.

3. Вычисляется вектор $\bar{y}_1$ по формуле

$$\bar{y}_1 = A \cdot \bar{w}_0. \quad (4.27)$$

На этом завершается одна итерация степенного метода.

В общем случае переход от $\bar{y}_k$ к вектору $\bar{y}_{k+1}$ совершается по формулам

$$d_k = \sqrt{\sum_{i=1}^n (y_{k,i})^2} \quad (4.28)$$

$$\bar{w}_k = 1/d_k \bar{y}_k \quad (4.29)$$

$$\bar{y}_{k+1} = A \cdot \bar{W}_k \quad (4.30)$$

Итерации проводятся до тех пор, пока не выполнится условие

$$|d_{k+1} - d_k| < \varepsilon.$$

Тогда можно принять, что $\bar{u}_1 \approx \bar{y}_k$, $\lambda_1 \approx d_k$. Это означает, что последовательность d_k сходится к максимальному собственному значению λ , а последовательность векторов $\bar{y}_k$ к собственному вектору $\bar{u}_1$, соответствующему значению λ .

Пример 14. Найти максимальное собственное значение и соответствующий собственный вектор для матрицы A

$$A = \begin{bmatrix} 10 & 1 \\ 1 & 2 \end{bmatrix}.$$

Решение. Задаем начальный вектор $\bar{y}_0 = \{1, 0\}$.

1 шаг. Вычисляем длину вектора $\bar{y}_0$

$$d_0 = \sqrt{1 + 0} = 1.$$

Для вектора $\bar{y}_0$ нормировка не требуется, так как он имеет единичную длину. Поэтому $\bar{W}_0 = \bar{y}_0$. Переходим к вычислению $\bar{y}_1 = A \cdot \bar{W}_0$

$$\bar{y}_1 = \begin{bmatrix} 10 & 1 \\ 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 10 \\ 1 \end{bmatrix}.$$

2 шаг. Вычисляем длину вектора $\bar{y}_1$

$$d_1 = \sqrt{10^2 + 1} = \sqrt{101} = 10,05.$$

Нормируем вектор $\bar{y}_1$, вычисляя $\bar{W}_1$

$$\bar{W}_1 = \frac{1}{d_1} \bar{y}_1 = \frac{1}{10,05} \begin{bmatrix} 10 \\ 1 \end{bmatrix} = \begin{bmatrix} 0,9950 \\ 0,0995 \end{bmatrix}.$$

Умножаем матрицу A на полученный вектор $\bar{W}_1$

$$\bar{y}_2 = A \cdot \bar{W}_1 = \begin{bmatrix} 10 & 1 \\ 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} 0,9950 \\ 0,0995 \end{bmatrix} = \begin{bmatrix} 10,05 \\ 1,194 \end{bmatrix}.$$

3 шаг. Вычисляем длину вектора $\bar{y}_2$

$$d_2 = \sqrt{10,05^2 + 0,0995^2} = \sqrt{102,428} = 10,1207.$$

Нормируем вектор $\bar{y}_2$, вычисляя $\bar{W}_2$

$$\bar{W}_2 = \frac{1}{10,1207} \begin{bmatrix} 10,05 \\ 1,194 \end{bmatrix} = \begin{bmatrix} 0,993 \\ 0,118 \end{bmatrix}.$$

Умножаем матрицу A на вектор $\bar{W}_2$

$$\bar{y}_3 = A \cdot \bar{W}_2 = \begin{bmatrix} 10 & 1 \\ 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} 0,993 \\ 0,118 \end{bmatrix} = \begin{bmatrix} 10,05 \\ 1,229 \end{bmatrix}.$$

4 шаг. Вычисляем длину вектора $\bar{y}_3$ и нормируем его

$$d_3 = \sqrt{10,05^2 + 1,229^2} = 10,125.$$

$$\bar{W}_3 = \frac{1}{10,125} \begin{bmatrix} 10,05 \\ 1,229 \end{bmatrix} = \begin{bmatrix} 0,9926 \\ 0,1214 \end{bmatrix}.$$

$$\bar{y}_4 = A \bar{W}_3 = \begin{bmatrix} 10 & 1 \\ 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} 0,9926 \\ 0,1214 \end{bmatrix} = \begin{bmatrix} 10,047 \\ 1,235 \end{bmatrix}$$

Видно, что величины d_2 и d_3 отличаются мало $|d_2 - d_3| = 0,0043$. Поэтому можно считать, что наибольшее собственное значение и соответствующий собственный вектор равны

$$\lambda_1 \approx d_3 = 10,125,$$

$$\bar{u}_1 \approx \bar{W}_3 = \begin{bmatrix} 0,9926 \\ 0,1214 \end{bmatrix}.$$

Для сравнения укажем, что максимальное собственное значение, полученное с помощью характеристического уравнения, равно $\lambda_{\max} = 10,123$, а $\lambda_{\min} = 1,887$.

Отметим, что скорость сходимости степенного метода зависит от того, насколько $\lambda_{\max}$ превышает остальные собственные значения. В рассмотренном выше примере $\lambda_{\max} / \lambda_{\min} \sim 5$ и поэтому степенной метод быстро сходится.

Фактически уже второй шаг дает хорошее приближение для λ_1 и $\bar{u}_1$.

Часто бывает нужно определить не наибольшее, а наименьшее собственное значение матрицы A . В этом случае удобно использовать соотношение (4.15), согласно которому собственные значения обратной матрицы A^{-1} обратно пропорциональны собственным значениям матрицы A . Отсюда следует, что

$$\lambda_{\min}(A) = 1/\lambda_{\max}(A^{-1}) \quad (4.31)$$

Это означает, что для нахождения минимального собственного значения матрицы A надо найти обратную матрицу A^{-1} , для нее найти максимальное собственное значение и воспользоваться формулой (4.31).

ГЛАВА 5. ОСОБЕННОСТИ РЕШЕНИЯ НЕЛИНЕЙНЫХ ЗАДАЧ НА ПРИМЕРЕ ОДНОГО УРАВНЕНИЯ

5.1. Некоторые определения

Здесь изучаются численные методы решения одного нелинейного уравнения (алгебраического или трансцендентного)*

$$F(x) = 0, \quad (5.1)$$

где $F(x)$ – непрерывная функция достаточно произвольного вида, определенная на некотором конечном или бесконечном интервале. Ограничения на ее свойства будут приведены при описании каждого из методов решения.

Корнем уравнения (5.1) называется всякое число x_* , обращающее в нуль функцию $F(x)$, то есть такое, что

$$F(x_*) = 0. \quad (5.2)$$

Задача нахождения всех корней уравнения (5.1) обладает следующими **характерными особенностями**:

1. В общем случае **не существует “конечных” алгоритмов** для получения корней уравнения (5.1), т.е. таких алгоритмов, которые позволяют за конечное число арифметических операций получить точное решение. Исключением являются уравнения типа

$$\sin x = 0, \quad (5.3)$$

* Напомним, что алгебраическими уравнениями называются уравнения, содержащие только алгебраические функции (целые, рациональные или иррациональные). Например, многочлен $a_0 + a_1x + a_2x^2 + \dots + a_nx^n$ является целой алгебраической функцией. Уравнения, содержащие другие функции (показательные, логарифмические, тригонометрические и др.), называются трансцендентными.

некоторые простые многочлены вида

$$ax(bx+c)=0, \quad (5.4)$$

и другие подбные уравнения, в которых решение может быть сразу выписано или без труда найдено с помощью небольшого числа арифметических операций. Даже в простом случае квадратного уравнения решение выписывается с помощью радикалов (квадратных корней), вычисление которых выполняется приближенно. Показано, что для многочленов выше четвертой степени точное решение не может быть получено даже с помощью радикалов. Поэтому обычно используются приближенные методы решения (5.1) – графические или численные.

Численные методы решения (5.1) являются, как правило, итерационными. Идея этих методов заключается в том, что, задавшись некоторым приближенным начальным значением x_0 для корня уравнения, можно с помощью некоторой совокупности операций найти уточненное значение корня. Выполнение этой совокупности операций определяет одну итерацию. Многократное повторение итераций, образующее итерационный процесс, позволяет получать все более точное значение корня (при условии, что итерационный процесс сходится). Таким образом, **итерационный процесс** заключается в вычислении последовательности значений $x_1, x_2, \dots, x_k, \dots$, где величина k указывает номер итерации или номер приближения. **Итерационный процесс сходится** к корню x уравнения, если $\lim x_k = x$ при $k \rightarrow \infty$.

2. Количество корней уравнения (5.1) обычно заранее неизвестно. При различных $F(x)$ уравнение (5.1) может вообще не иметь корней, иметь один, несколько или даже бесчисленное множество корней. Например, уравнение $x^2 + 5 = 0$ не имеет действительных корней; уравнение $e^x + x = 0$ имеет единственный корень; уравнение (5.4) – два корня (при $b \neq 0$), уравнение (5.3) – бесчисленное множество корней. Поэтому численному решению (5.1) должен предшествовать анализ уравнения и выяснение того, сколько корней оно имеет. Это удобнее всего сделать методами математического анализа. Кроме того, необходимо **отделить все корни**, т.е. для каждого корня x указать такой интервал $[a, b]$, в котором расположен данный корень ($x \in [a, b]$) и других корней в нем нет. При использовании итерационных методов предполагается, что отделение корней проведено, так что итерационный процесс используется для приближенного определения единственного корня на заданном интервале.

3. Сходимость большинства итерационных методов зависит от начального приближения x_0 . Как правило, сходимость гарантируется только в том случае, если x_0 достаточно близко к корню уравнения x . Иногда полезно использовать понятие области притяжения корня. **Областью притяжения корня x** для данного итерационного метода называется такое множество A значений величины x_0 , что для любого $x_0 \in A$ итерационный метод сходится к корню x .

При сравнении сходимости различных методов часто используется понятие **порядка сходимости метода**. Говорят, что метод имеет n -ый порядок сходимости, если

$$|x_k - x_*| \leq A \cdot |x_{k-1} - x_*|^n, \quad (5.5)$$

где A – константа, зависящая от вида функции $F(x)$, но не зависящая от номера итерации k . Неравенство (5.5) фактически связывает погрешности на двух последовательных итерациях $\varepsilon_k = |x_k - x_*|$ и $\varepsilon_{k-1} = |x_{k-1} - x_*|$ и показывает, насколько быстро сходится итерационный процесс. Понятно, что чем больше величина n , тем быстрее он сходится. Ниже будут рассмотрены методы, имеющие линейную сходимость

$$|x_k - x_*| \leq A \cdot |x_{k-1} - x_*| \quad (5.6)$$

и квадратичную сходимость

$$|x_k - x_*| \leq A \cdot |x_{k-1} - x_*|^2 \quad (5.7)$$

Величина A для каждого метода различна и зависит, кроме того, от вида функции $F(x)$.

Рассмотрим пример отделения корней.

Пример 15. Определить количество корней уравнения $e^x - x - 2 = 0$ и отделить корни.

Решение. Запишем уравнение в виде $e^x = x + 2$ и построим графики функций, стоящих в левой и правой части уравнения $y_1 = e^x$ и $y_2 = x + 2$ (рис. 5.1).

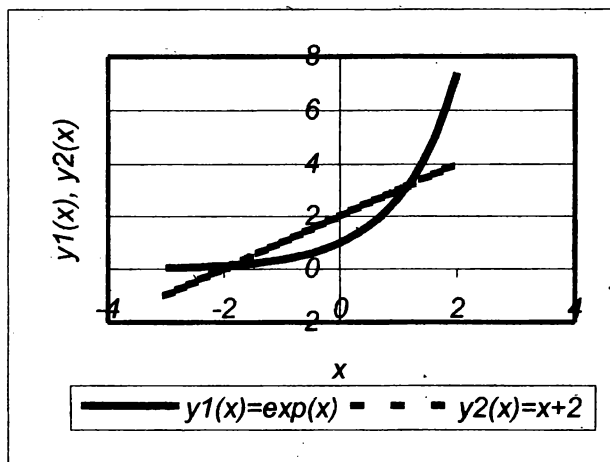


Рис. 5.1

Корнями уравнения будут те значения x , при которых эти графики пересекаются. При $x = 1$ график прямой лежит выше графика функции $y_1 = e^x$, так как при $x = 1$ $y_1 = e^1 = e \approx 2,71$, а $y_2 = 1 + 2 = 3$. При $x = 2$ прямая расположена ниже графика $y_1 = e^x$, так как $e^2 \approx 7,4 > 2 + 2 = 4$. Следовательно, один из корней находится в интервале $[1, 2]$. Вычислив значения функций $y_1 = e^x$ и $y_2 = x + 2$ при $x = -2$ и $x = -1$, определяем, что второй корень (отрицательный) находится внутри интервала $[-2, -1]$. Зная поведение функций $y_1 = e^x$ и $y_2 = x + 2$, мы из рис. 5.1 можем заключить, что других корней у рассматриваемого уравнения нет. Функция $y_1 = e^x$ с ростом x растет гораздо быстрее, чем прямая $y_2 = x + 2$ и при $x > 2$ других корней быть не может. Аналогично обстоит дело при $x < -2$, так как $e^x \rightarrow 0$ при $x \rightarrow -\infty$, а $y_2 = x + 2 \rightarrow -\infty$ при $x \rightarrow -\infty$. Из этого рассмотрения следует, что положительный корень уравнения надо искать на интервале $[1, 2]$, а отрицательный – на интервале $[-2, -1]$, причем положительный корень ближе к точке $x = 1$, а отрицательный – к точке $x = -2$.

5.2. Численное нахождение корня на ЭВМ

Специфические вопросы возникают при численном нахождении корней уравнений на ЭВМ. Во-первых, требуется уточнить понятие корня с точки зрения “машинной” математики. Данное выше определение корня уравнения (5.1) может иногда не иметь смысла просто потому, что такого числа x среди чисел, которые могут быть представлены в ЭВМ, нет. Например, простое уравнение $x^2 - 2 = 0$ имеет корни $x_1 = \sqrt{2}$ и $x_2 = -\sqrt{2}$, которые будучи иррациональными числами, не могут быть точно представлены в ЭВМ.* Поэтому с практической точки зрения важно решить, что понимается под приближенным значением корня. Фактически это вопрос о том, когда надо прекратить итерационный процесс нахождения корня и в каком смысле значение x_k можно приближенно считать корнем уравнения. Один из разумных подходов заключается в том, что указывается малый интервал $[\alpha, \beta]$, в котором находится корень. Длина этого интервала $\beta - \alpha$ определяет **погрешность нахождения корня**

$$\varepsilon_k = |x_k - x_*| \leq \beta - \alpha \quad (5.8)$$

На таком подходе основан, например, метод половинного деления, который будет рассмотрен ниже.

Иногда величину полученной погрешности ε_k можно оценить, зная поведение производной $F'(x)$ в окрестности корня. Пусть x_* – точное, а x_k – приближенное значение корня, они оба лежат в интервале $[a, b]$, а функция $F'(x)$ имеет на этом интервале минимум, т.е. $|F'(x)| \geq m$ при $x \in [a, b]$. Тогда по теореме Лагранжа

$$F(x_k) - F(x_*) = (x_k - x_*) \cdot F'(\xi), \quad (5.9)$$

* Иррациональные числа являются бесконечными непериодическими дробями, а в ЭВМ любое число можно представить лишь конечным числом разрядов.

где ξ лежит в промежутке между значениями x_k и x_* и, следовательно, $\xi \in [a, b]$. Учитывая, что $F(x_*) = 0$, из (5.9) получим оценку погрешности для приближенного значения корня

$$|x_k - x_*| \leq \frac{1}{m} F(x_k). \quad (5.10)$$

Величину $F(x_k)$ называют невязкой уравнения (5.1). Иногда в качестве условия окончания итерационного процесса принимается, что

$$|F(x_k)| < \varepsilon. \quad (5.11)$$

Неравенство (5.10) показывает, что использование условия (5.11) может привести к неверным результатам, когда производная достаточно мала. На рис. 5.2 видно, что хотя $|F(x_k)| < \varepsilon$, приближенное значение x_k сильно отличается от корня. Впрочем, оценка точности корня по (5.11) является в принципе ненадежной, так как, умножив уравнение (5.1) на любую константу, мы получим равносильное уравнение $AF(x) = 0$ и, следовательно, невязка уравнения $|AF(x_k)|$ определяется величиной константы A , а не близостью приближенного значения к точному.

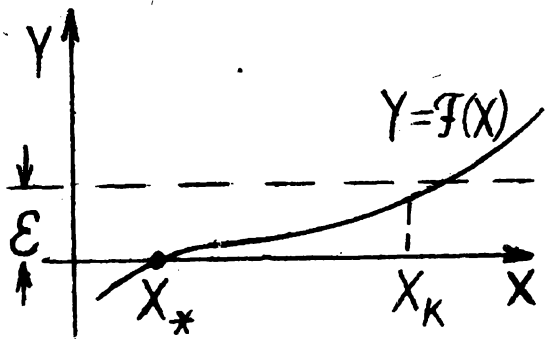


Рис. 5.2

Другим условием, применяемым для окончания итерационного процесса, является близость двух последовательных приближений

$$|x_{k+1} - x_k| < \varepsilon. \quad (5.12)$$

Вообще говоря, из малости $|x_{k+1} - x_k|$ при конечных значениях k не следует малость погрешности $|x_k - x_*|$. Поэтому при рассмотрении различных

итерационных методов необходимо получить не только условие сходимости, но и, по возможности, оценки погрешности решения, позволяющие определить их величину за конечное число итераций. В этом смысле очень удобными являются так называемые двусторонние приближения и итерационные процессы с немонотонной сходимостью.

Под **двусторонним приближением** понимается нахождение корня двумя различными приближенными методами, один из которых дает завышенное значение корня, а другой – заниженное. В этом случае на каждом итерационном шаге имеется интервал, в котором находится корень. Длина интервала ограничивает погрешность вычисления корня.

Итерационный процесс называется **монотонно сходящимся**, если величина погрешности $x_k - x_*$ сохраняет одинаковый знак при рассматриваемых значениях k . Процесс называется **колеблющимся**, если $(x_{k+1} - x_*)(x_k - x_*) < 0$ (приближенные значения оказываются то больше, то меньше точного значения x_*). Для колеблющихся итерационных процессов на каждом шаге имеется оценка погрешности вычисления корня

$$|x_{k+1} - x_*| \leq |x_{k+1} - x_k|. \quad (5.13)$$

Очевидно, что двусторонние приближения и колеблющиеся итерационные процессы очень удобны в практических расчетах, так как они на каждом шаге дают надежные границы для значения корня.

Перейдем к рассмотрению различных итерационных методов.

5.3. Метод половинного деления и метод хорд

Метод половинного деления – всегда сходящийся для непрерывной функции метод нахождения корня.

Предположим, что корни уравнения (5.1) уже отделены и на заданном интервале $[a, b]$ уравнение имеет единственный корень x_* , и, следовательно, на концах интервала функция имеет разные знаки: $F(a)F(b) < 0$. Никаких ограничений на функцию $F(x)$, кроме требования непрерывности, не накладывается. Идея метода заключается в том, чтобы вместо исходного интервала $[a, b]$ получить путем последовательного деления отрезка пополам достаточно малый интервал $[\alpha, \beta]$, внутри которого находится корень.

Разделим интервал $[a, b]$ пополам, вычислив координату середины отрезка $c_0 = (a+b)/2$. В результате получим два интервала: $[a, c_0]$ и $[c_0, b]$. Если окажется, что $F(c_0) = 0$, то значит c_0 является корнем ($x_* = c_0$) и, следовательно, решение на этом заканчивается. Если же $F(c_0) \neq 0$, то из двух интервалов выбираем тот, на концах которого функция $F(x)$ имеет разные знаки и внутри которого, следовательно, находится корень. Второй интервал, на котором нет корня, отбрасывается. Для функции $F(x)$, приведенной на рисунке 5.3, $F(c_0)F(a) > 0$ и поэтому $F(c_0)F(b) < 0$, откуда следует, что надо оставить отрезок $[c_0, b]$, а отрезок $[a, c_0]$ отбросить, так как $x_* \in [c_0, b]$.

Если обозначить координаты оставленного интервала через a_1 и b_1 , то мы получим, что в общем случае $a_1 = a$, $b_1 = c_0$, если $F(a)F(c_0) < 0$, и $a_1 = c_0$, $b_1 = b$, если $F(a)F(c_0) > 0$. Длина полученного интервала $[a_1, b_1]$, внутри которого

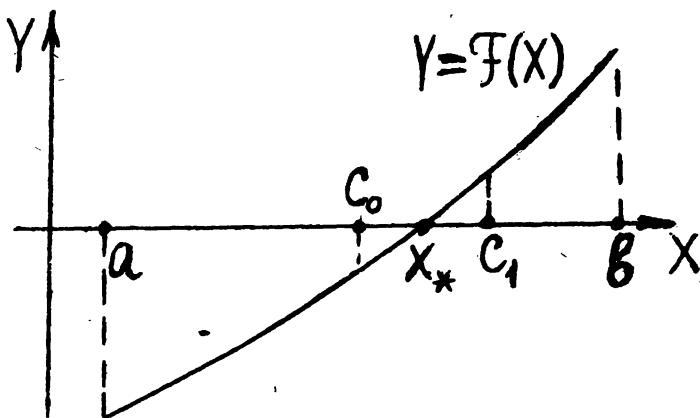


Рис 5.3

находится корень x_* , будет вдвое меньше длины $b - a$ исходного интервала. На этом заканчивается один шаг метода. Разделим теперь полученный интервал $[a_1, b_1]$ снова пополам, вычислив $c_1 = (a_1 + b_1)/2$ и из полученных двух интервалов $[a_1, c_1]$ и $[c_1, b_1]$ возьмем тот, на концах которого $F(x)$ имеет разные знаки. Продолжая этот процесс, мы получим последовательность вложенных интервалов $[a_1, b_1], [a_2, b_2], \dots, [a_k, b_k], \dots$, внутри которых находится корень и длина которых монотонно убывает, так как

$$b_k - a_k = \frac{1}{2^k} (b - a). \quad (5.14)$$

Процесс вычислений прекращается, когда $b_k - a_k < \varepsilon$, где ε – требуемая точность решения задачи. Таким образом, в методе половинного деления находится не значение корня, а малый интервал $[a_k, b_k]$, в котором находится этот корень, что, учитывая особенность вычислений на ЭВМ, полностью решает задачу (см. раздел 5.2).

Главное достоинство метода заключается в том, что он сходится всегда, независимо от того, как ведут себя функция $F(x)$ и ее производные. Единственное требование – наличие разных знаков на концах интервала (и, разумеется, непрерывность $F(x)$). Если на интервале имеется не единственный корень, то метод позволяет получить один из корней. Недостатком метода является сравнительно медленная сходимость. Так как на каждом шаге интервал уменьшается вдвое, то справедливо соотношение (5.14), откуда видно, что через 10 шагов метода длина исходного интервала $[a, b]$ составит $(1/2)^{10} \cdot (b - a) \approx 0,001(b - a)$, а через 20 шагов – $10^{-6}(b - a)$. В дальнейшем будут описаны другие методы, некоторые из которых сходятся к решению значительно быстрее, но только в том случае, если начальное приближение x_0 достаточно близко к корню x_* , и если функция $F(x)$ и ее производные удовлетворяют некоторым условиям.

Пример 16. Найти методом половинного деления положительный корень уравнения $f(x) \equiv e^x - x - 2 = 0$ с точностью $\varepsilon = 0,02$.

Решение. В примере 15 было показано, что этот корень находится внутри интервала $[1, 2]$. Приведем полученные на различных шагах метода результаты в виде таблицы 5.1.

Таблица 5.1

k	a_k	b_k	c_k	$f(a_k)$	$f(b_k)$	$f(c_k)$	$b_k - a_k$
0	1	2	1,5	-0,2817	3,3891	0,9817	1
1	1	1,5	1,25	-0,2817	0,9817	0,2403	0,5
2	1	1,25	1,125	-0,2817	0,2403	-0,0448	0,25
3	1,125	1,25	1,1875	-0,0448	0,2403	0,0914	0,125
4	1,125	1,1875	1,15625	-0,0448	0,0914	0,0217	0,0625
5	1,125	1,15625	1,1406	-0,0448	0,0217	-0,0188	0,03125
6	1,1406	1,15625					0,01565

Ответ: $x_k \in [1,1406; 1,15625]$

Используя основную идею метода половинного деления (разбиение исходного отрезка на два и отбрасывание того отрезка, где нет корня), можно модифицировать метод, деля отрезок не пополам, а в каком либо другом отношении. Одной из таких модификаций является метод хорд, существо которого заключается в следующем (рис. 5.4, 5.5).

В качестве точки c_0 , делящей отрезок $[a, b]$ на две части, выбирается точка пересечения с осью x прямой AB , проведенной через две точки $A(a, F(a))$ и $B(b, F(b))$. Получив два отрезка $[a, c_0]$ и $[c_0, b]$, мы оставляем тот из них, на концах которого функция $F(x)$ имеет разные знаки и на котором, следовательно, находится корень. На рисунке 5.4 это отрезок $[a, c_0]$, для которого $F(a) \cdot F(c_0) < 0$, а на рисунке 5.5 – отрезок $[c_0, b]$, для которого $F(c_0) \cdot F(b) < 0$. Прямая AB будет хордой, соединяющей две крайние точки кривой $F(x)$. На следующем шаге метода проводится хорда через крайние точки функции $F(x)$ на отрезке $[a, c_0]$ (или $[c_0, b]$ на рис. 5.5) и определяется точка c_1 пересечения этой хорды с осью x . Для функции $F(x)$, приведенной на рис. 5.4, видно, что все приближения метода хорд $c_0, c_1, c_2, \dots$ лежат правее корня x ($c_k > x$) и итерационный процесс будет монотонным. Аналогично обстоит дело для функции, приведенной на рис. 5.5, но теперь последовательность $c_0, c_1, c_2, \dots, c_k, \dots$ будет лежать левее корня. Итерационный процесс продолжается до тех пор, пока значение $F(c_k)$ не станет по модулю меньше заданного малого числа ε . Значения величин $c_0, c_1, c_2, \dots, c_k$ определяются следующим образом. Уравнение прямой AB , проходящей через точки $A(a, F(a))$ и $B(b, F(b))$ имеет вид

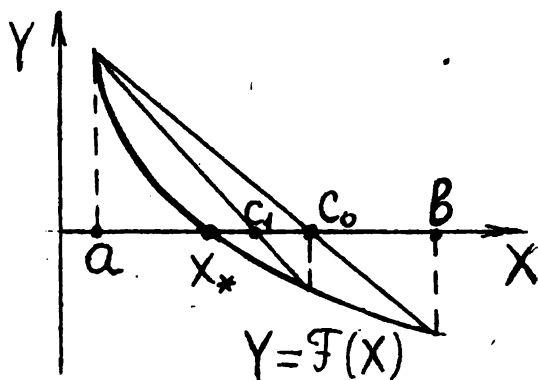


Рис. 5.4

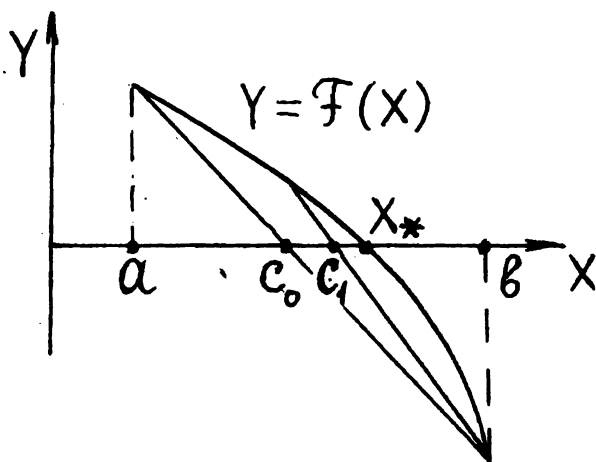


Рис. 5.5

$$\frac{x - a}{b - a} = \frac{y - F(a)}{F(b) - F(a)}$$

Для точки ее пересечения с осью x ($x = c_0$, $y = 0$) получим уравнение

$$\frac{c_0 - a}{b - a} = \frac{-F(a)}{F(b) - F(a)}$$

откуда для c_0 получим выражение

$$c_0 = a - \frac{b-a}{F(b)-F(a)} F(a). \quad (5.15)$$

Аналогично определяется следующее приближение c_1 как пересечение с осью x хорды для $F(x)$ на отрезке $[a, c_0]$ ([c_0, b]) для случая на рис. 5.5). Доказательство сходимости метода хорд приведено, например, в [9].

Для функции $F(x)$, приведенной на рис. 5.4 и 5.5, первая и вторая производные $F'(x)$ и $F''(x)$ не обращаются в нуль ни в одной точке интервала $[a, b]$. В этом случае сходимость метода хорд будет монотонной. На самом деле применение метода хорд в том виде, как он здесь описан, не накладывает никаких ограничений на функцию $F(x)$, кроме ее непрерывности и условия $F(a)F(b) < 0$, достаточных для существования хотя бы одного корня на отрезке $[a, b]$. Как и метод половинного деления, **метод хорд сходится всегда**, однако в ряде случаев он дает более быструю сходимость итерационного процесса, чем метод половинного деления. Однако возможны случаи, когда он сходится очень медленно. Для иллюстрации на рис. 5.6 приведена сходимость для случая, когда $F'(x) = 0$ и $F''(x) = 0$ внутри интервала $[a, b]$, а на рис.5.7 приведен случай медленной сходимости метода хорд.

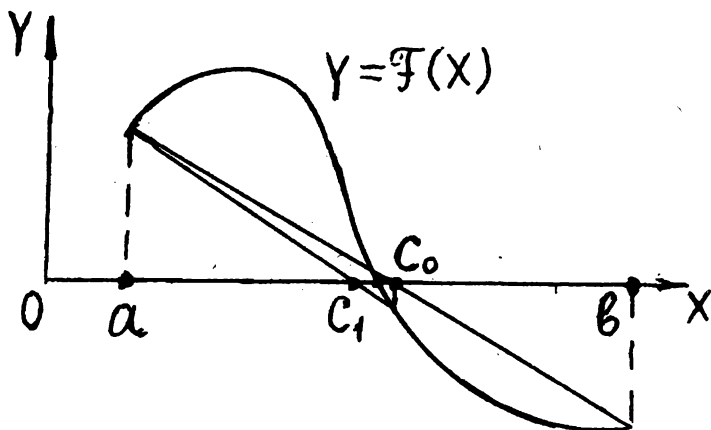


Рис. 5.6

Сходимость метода хорд в случае, приведенном на рис. 5.6, не будет, конечно, монотонной, как это было на рис. 5.4 и 5.5. Геометрически очевидно, что для случая, приведенного на рис. 5.7, метод половинного деления будет сходиться гораздо быстрее, чем метод хорд. Такая ситуация является типичной для нелинейных задач. **Сходимость и сравнительные достоинства различных методов могут существенно зависеть от поведения функции $F(x)$ и ее производных.**

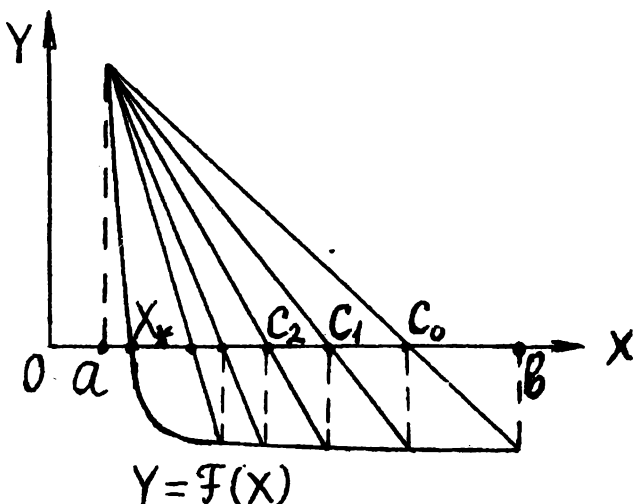


Рис. 5.7

Завершая рассмотрение двух всегда сходящихся методов, отметим одно несомненное достоинство метода половинного деления: для него всегда заранее можно сказать, сколько шагов потребуется для получения корня с заданной точностью (независимо от вида функции $F(x)$). Эта оценка делается на основании формулы (5.14). Метод хорд таким свойством не обладает, так как скорость его сходимости зависит от вида функции $F(x)$.

Пример 17. Найти методом хорд положительный корень уравнения $F(x) \equiv e^x - x - 2 = 0$ с точностью $\varepsilon = 0,01$.

Решение. В примере 15 было показано, что этот корень находится в интервале $[1, 2]$. Примем $a = 1$, $b = 2$ и проведем несколько итераций методом хорд по формуле (5.15). Результаты расчетов приведены в таблице 5.2. Сравнение этих результатов с результатами метода половинного деления (таблица 5.1) показывает, что для уравнения $e^x - x - 2 = 0$ метод хорд сходится быстрее.

Таблица 5.2

k	a_k	b_k	c_k	$F(a_k)$	$F(b_k)$	$F(c_k)$	интервал	$ c_k - c_{k-1} $
0	1	2	1,6216	-0,2817	3,3891	1,4395	$[a, b]$	
1	1	1,6216	1,1017	-0,2817	1,4395	-0,09236	$[a, b_1]$	0,5199
2	1,1017	1,6216	1,1331	-0,0924	1,4395	-0,0280	$[c_1, b_2]$	0,0321
3	1,1331	1,6216	1,1424	-0,0280	1,4395	-0,00822	$[c_2, b_3]$	0,0093

5.4. Метод простой итерации

5.4.1. Описание метода и условия его сходимости

Рассмотрим уравнение (5.1)

$$F(x) = 0.$$

Для того, чтобы применить метод простой итерации, необходимо прежде всего привести уравнение (5.1) к виду, удобному для проведения итерационного процесса

$$x = \varphi(x), \quad (5.16)$$

причем уравнения (5.16) и (5.1) должны быть равносильны, т.е. все корни одного уравнения должны быть корнями другого уравнения, и наоборот. Привести (5.1) к виду (5.16) можно различными способами. Рассмотрим пример.

Пример 18. Привести уравнение

$$x^3 + x - 0,5 = 0. \quad (5.17)$$

к виду, удобному для итераций (5.16).

Решение. Если перенести все члены, кроме второго, в правую часть, получится равносильное уравнение

$$x = 0,5 - x^3. \quad (5.18)$$

Если перенести второй и третий члены в правую часть и извлечь из обеих частей кубический корень, получится уравнение

$$x = \sqrt[3]{0,5 - x}. \quad (5.19)$$

Наконец, умножив обе части (5.17) на некоторое действительное число λ и прибавив к обеим частям уравнения величину x , получим

$$x = x + \lambda \cdot (x^3 + x - 0,5). \quad (5.20)$$

Уравнения (5.18) – (5.20) равносильны (5.17) и имеют вид (5.16), где в качестве $\varphi(x)$ будут соответственно $\varphi_1(x) \equiv 0,5 - x^3$, $\varphi_2(x) \equiv \sqrt[3]{0,5 - x}$, $\varphi_3(x) \equiv x + \lambda \cdot (x^3 + x - 0,5)$. Можно указать еще множество других способов сведения уравнения (5.17) к виду (5.16). Позже будет показано, что сходимость или расходимость итерационного процесса зависит от того, какой из этих способов будет использован. Отметим, что в (5.20) имеется свободный параметр λ , выбирая конкретные значения которого, можно обеспечить сходимость итерационного процесса. Более того, изменяя величину λ , можно управлять скоростью сходимости.

Метод простой итерации для уравнения $x = \varphi(x)$ заключается в следующем. Зададимся начальным приближением x_0 и построим последовательность чисел $x_1, x_2, \dots, x_k, \dots$ с помощью формул

$$x_1 = \varphi(x_0), x_2 = \varphi(x_1), \dots, x_k = \varphi(x_{k-1}), \dots \quad (5.21)$$

Если эта последовательность сходится, то для непрерывной функции $\varphi(x)$ она сходится к корню уравнения (5.16). Действительно, пусть $\lim x_k = d$ при $k \rightarrow \infty$. Тогда

$$d = \lim x_k = \lim \varphi(x_{k-1}) = \varphi(\lim x_{k-1}) = \varphi(d)$$

Отсюда следует, что d – корень уравнения (5.16).

Достаточное условие сходимости метода простой итерации определяется следующей теоремой.

Теорема. Пусть x_* – корень уравнения (5.16), т.е.

$$x_* = \varphi(x_*) \quad (5.22)$$

Пусть функция $\varphi(x)$ определена и дифференцируема на отрезке $[a, b]$ и все элементы последовательности (5.21) также принадлежат этому отрезку, т.е. $(x_k \in [a, b])$. Пусть существует действительное число q такое, что

$$|\varphi'(x)| \leq q < 1 \quad (x \in [a, b]) \quad (5.23)$$

Тогда процесс итераций (5.21) сходится и предельное значение $x_* = \lim x_k$ ($k \rightarrow \infty$) является единственным корнем уравнения (5.16).

Приведем доказательство теоремы, так как по ходу дела будет получена оценка скорости сходимости метода.

Доказательство. Оценим по формуле Лагранжа погрешность на двух последовательных итерациях. В силу (5.21) и (5.22) и ввиду того, что $x_k \in [a, b]$,

$$x_k - x_* = \varphi(x_{k-1}) - \varphi(x_*) = \varphi'(\zeta) \cdot (x_{k-1} - x_*) \quad (\zeta \in [x_k, x_*])$$

откуда в силу (5.23)

$$|x_k - x_*| \leq q \cdot |x_{k-1} - x_*| \quad (5.24)$$

Применяя (5.24) для $k = 1$, получим

$$|x_1 - x_*| \leq q \cdot |x_0 - x_*|$$

Для $k = 2$, используя предыдущее неравенство, имеем

$$|x_2 - x_*| \leq q \cdot |x_1 - x_*| \leq q^2 |x_0 - x_*|$$

С помощью метода математической индукции легко показать, что в общем случае

$$|x_k - x_*| \leq q^k \cdot |x_0 - x_*| \quad (5.25)$$

Так как $q < 1$, то из (5.25) следует, что последовательность x_k сходится к

корню уравнения $x_k \rightarrow x$. ($k \rightarrow \infty$), так как $q^k \rightarrow 0$ (при $k \rightarrow \infty$). Скорость этой сходимости не хуже, чем у геометрической прогрессии q^k ($q < 1$). Покажем, что корень x является единственным. Действительно, пусть существует другой корень $\bar{x}$, так что $\bar{x} = \varphi(\bar{x})$, $\bar{x} \in [a, b]$. Тогда по формуле Лагранжа $x_* - \bar{x} = \varphi(x_*) - \varphi(\bar{x}) = \varphi'(\zeta) \cdot (x_* - \bar{x})$, где $\zeta \in [x_*, \bar{x}]$, откуда

$$|x_* - \bar{x}| \leq q \cdot |x_* - \bar{x}| < |x_* - \bar{x}|$$

Полученное противоречие доказывает единственность корня.

Замечания.

1. При доказательстве теоремы использовалось предположение о том, что последовательность x_k не выходит за пределы интервала $[a, b]$, так как в противном случае для значений этой последовательности нельзя было бы использовать неравенство (5.23). Можно показать, что если x_0 принадлежит окрестности корня x , т.е. $|x_0 - x| < \delta$ и вся δ -окрестность корня находится в интервале $[a, b]$, то все приближения x_k будут оставаться в этой окрестности.
2. Из (5.25) следует, что чем меньше величина q , тем быстрее сходится итерационный процесс.
3. Неравенство (5.24) означает, что метод простой итерации имеет первый порядок сходимости (см. формулу (5.5), определяющую порядок сходимости любого метода).

5.4.2. Принцип сжатых отображений

Выше была доказана теорема о сходимости метода итераций. Эта теорема является частным случаем весьма общей теоремы функционального анализа о сжатых отображениях, которая очень широко используется в вычислительной математике. Поскольку основы функционального анализа студенты будут изучать позже, приведем здесь существо этой теоремы, не вводя аккуратно необходимые понятия функционального анализа.

Пусть X и Y два множества. Будем говорить, что оператор $y = \Phi x$ отображает X в Y , если каждому элементу $x \in X$ поставлен в соответствие элемент $y \in Y$. Если $X = Y$, то этот оператор задает отображение Φ пространства X в себя. Всякая точка x , которая переводится отображением Φ в себя (т.е. для которой $\Phi x = x$), называется неподвижной точкой этого отображения. Пусть на множествах X и Y каким-либо образом задано расстояние между двумя любыми элементами x_1 и x_2 (или y_1 и y_2), которое мы обозначим как $\rho(x_1, x_2)$ или $\rho(y_1, y_2)$.

Отображение $y = \Phi x$, удовлетворяющее условию

$$\rho(y_1, y_2) \leq q \rho(x_1, x_2) \quad (5.26)$$

для любых $x_1, x_2 \in X$ при $0 < q < 1$, называется **сжатым**. Смысл этого понятия заключается в том, что расстояние между любыми точками y_1 и y_2

$(y_1, y_2 \in Y)$, получаемыми с помощью преобразования $y = \Phi x$ из точек x_1 и x_2 , будет меньше расстояния между точками x_1 и x_2 (отображение "сжимает" расстояние между точками).

Теорема (принцип сжатых отображений). Если отображение $y = \Phi x$ сжатое, то существует одна и только одна неподвижная точка этого отображения, т.е. уравнение $x = \Phi x$ имеет единственное решение. Решение этого уравнения может быть получено как предел последовательности

$$\varphi^{(k)} = \Phi x^{(k-1)} \quad k = 1, 2, 3, \dots, \quad (5.27)$$

где $x^{(k)}$ – произвольный элемент, $x^{(k)} \in X$.

Для функции одной переменной $y = \varphi(x)$ отображение $\varphi(x)$ будет сжатым, если

$$|\varphi(x_2) - \varphi(x_1)| \leq q |x_2 - x_1| \quad (q < 1) \quad (5.28)$$

Рассматривая итерационный процесс $x_k = \varphi(x_{k-1})$, можно заметить, что если для $\varphi(x)$ выполняется (5.28), то выполняется и

$$|\varphi(x_{k-1}) - \varphi(x_*)| \leq q |x_{k-1} - x_*| \quad (q < 1) \quad (5.29)$$

где x_* – корень уравнения $x = \varphi(x)$. Отсюда получаем, что

$$|x_k - x_*| \leq q \cdot |x_{k-1} - x_*|. \quad (5.30)$$

(5.30) означает сходимость итерационного процесса $x_k = \varphi(x_{k-1})$ со скоростью не меньшей, чем сходимость геометрической прогрессии при $q < 1$. Отметим, что для выполнения условия (5.28) достаточно, например, чтобы выполнялось условие (5.23).

В качестве другого примера применения принципа сжатых отображений укажем, что условие сходимости метода Зейделя для систем линейных уравнений, рассмотренное в разделе 3.4, также является следствием принципа сжатых отображений для линейной системы, записанной в виде, удобном для итераций.

5.4.3. Геометрический смысл метода итерации

Геометрический смысл метода итераций иллюстрируется на рис.5.8, 5.9.

Корень x_* является абсциссой точки пересечения прямой $y = x$ и кривой $y = \varphi(x)$. Точка x_1 геометрически находится из условия, что $x_1 = |OA| = |AB| = \varphi(x_0)$. Геометрически метод итераций приводит к построению ломаной (на рис. 5.8 и 5.9 сплошная черта), звенья которой параллельны осям x и y . Следует отметить, что в случае, когда $0 < \varphi'(x) < 1$ в окрестности корня, сходимость метода является монотонной: все значения x_k находятся по одну сторону от корня x_* , причем для всех k $x_{k+1} > x_k$, если $x_0 < x_*$, и $x_{k+1} < x_k$, если $x_0 > x_*$ (рис. 5.8). Если же $-1 < \varphi'(x) < 0$, как это, имеет место на рис. 5.9, то сходимость будет колеблющейся: значения x_k

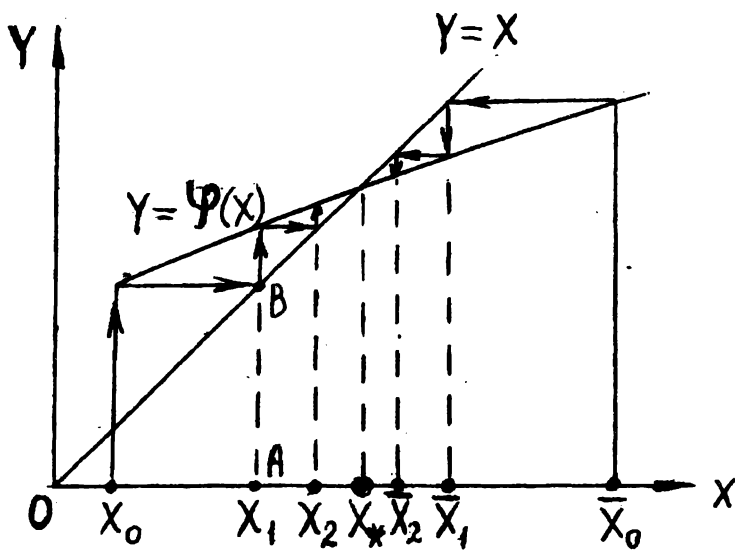


Рис. 5.8

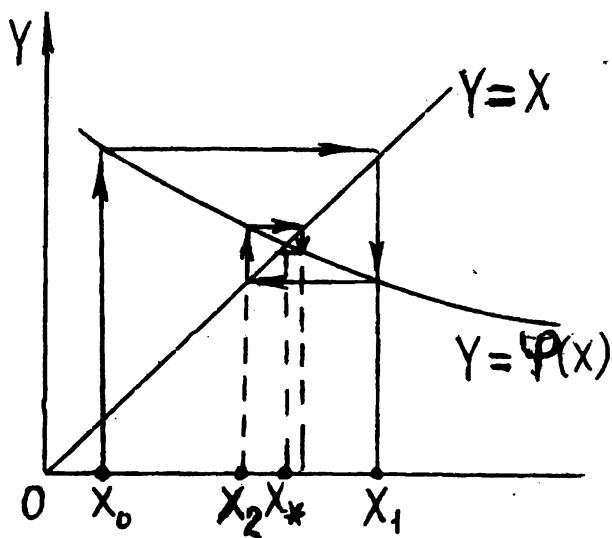


Рис. 5.9

будут поочередно то больше, то меньше значения корня x . (рис. 5.9), так как для всех k будет выполняться неравенство

$$(x_{k+1} - x_*)(x_k - x_*) < 0.$$

На практике колеблющаяся сходимость удобна тем, что любые два последовательных приближения x_k и x_{k+1} указывают интервал, в котором находится корень, и, следовательно, дают оценку погрешности вычисления приближенного значения корня

$$|x_{k+1} - x_*| \leq |x_{k+1} - x_k|$$

5.4.4. Примеры применения метода простой итерации

Пример 19. Исследовать применение метода итераций на примере уравнения (5.17)

$$x^3 + x - 0,5 = 0.$$

Решение. Построив функцию $y = x^3$ и $y = 0,5 - x$ графически, легко определить, что корень находится в интервале $0,3 \leq x \leq 0,5$. Как было показано выше (пример 18 в разделе 5.4.1), привести уравнение (5.17) к виду (5.16), удобному для итераций, можно различными способами (см. уравнения (5.18), (5.19), (5.20)). Ниже приведены эти уравнения вместе с вычисленными производными от правых частей

$$x = 0,5 - x^3 \equiv \varphi_1(x) \quad \varphi_1'(x) = -3x^2 \quad (5.31)$$

$$x = \sqrt[3]{0,5 - x} \equiv \varphi_2(x) \quad \varphi_2'(x) = \frac{1}{3} \cdot \frac{1}{\sqrt[3]{(0,5 - x)^2}} \quad (5.32)$$

$$x = x + \lambda(x^3 + x - 0,5) \equiv \varphi_3(x) \quad \varphi_3'(x) = 1 + 3\lambda x^2 + \lambda \quad (5.33)$$

Из (5.31) и (5.32) видно, что при $x_k \in [0,3; 0,5]$ $|\varphi_1'(x_k)| < 1$, а $|\varphi_2'(x_k)| > 1$. Следовательно, метод простой итерации для (5.31) сходится, а для (5.32) – расходится, если взять $x_0 \in [0,3; 0,5]$.

Свободный параметр λ в (5.33) можно использовать для того, чтобы итерационный процесс (5.33) сходиллся в рассматриваемом интервале, т.е. чтобы выполнялось неравенство $|\varphi_3'(x)| < 1$. При $x_k = 0,5$ получим неравенство $|1 + 1,75 \cdot \lambda| < 1$, откуда следует, что для сходимости значение λ должно удовлетворять неравенству

$$-1,14 < \lambda < 0. \quad (5.34)$$

Для другого конца интервала $x_k = 0,3$ получим

$$-1,57 < \lambda < 0$$

Из этих неравенств следует, что для любых $x_k \in [0,3; 0,5]$ метод итерации (5.33) сходится, если λ удовлетворяет условиям (5.34).

Пользуясь произвольностью значения λ , можно выбрать такое значение λ_0 из интервала (5.34), чтобы итерационный процесс сходился как можно быстрее. Неравенство (5.25) показывает, что итерационный процесс сходится тем быстрее, чем меньше величина q (напомним, что $q = \max|\varphi'(x)|$ для рассматриваемого интервала). Выберем λ_0 таким образом, чтобы $\varphi'_3(x) \equiv 1 + 3\lambda_0 x^2 + \lambda = 0$, откуда

$$\lambda_0 = -\frac{1}{3x^2 + 1} \quad (5.35)$$

При $x_0 = 0,5$ $\lambda_0 = -0,57143$. Результаты расчета корня уравнения (5.17) с помощью двух различных реализаций метода простой итерации (5.31) и (5.33), полученные на различных итерационных шагах, приведены в таблице 5.3. Точное значение $x = 0,4238538$.

Таблица 5.3

Итерации по (5.31)			Итерации по (5.33) $\lambda_0 = -0,57143$		
k	x_k	$ x_k - x_* $	k	x_k	$ x_k - x_* $
0	0,5	0,076146	0	0,5	0,076146
3	0,410526	0,013328	2	0,4244065	0,002448
6	0,425890	0,002036	3	0,4239202	0,000066
9	0,423534	0,000320	4	0,4238618	0,000008
12	0,423904	0,000050	5	0,4238548	0,000001
15	0,423847	0,000007	6	0,4238539	0,0000001

Результаты, приведенные в таблице, показывают, что итерационный процесс (5.33) с выбором λ_0 по формуле (5.35) сходится гораздо быстрее, чем метод (5.31). Как мы увидим позже, метод (5.33) очень близок к методу Ньютона, который в отличие от метода простой итерации имеет второй порядок сходимости.

5.5. Метод Ньютона

5.5.1. Общее описание и геометрический смысл

Метод Ньютона (касательных) – быстро сходящийся итерационный метод нахождения корня, сходимость которого сильно зависит от выбора начального приближения

Рассмотрим уравнение (5.1)

$$F(x) = 0$$

Пусть x_* – корень уравнения (5.1), $x_* \in [a, b]$. Пусть у $F(x)$ существует первая и вторая производные $F'(x)$ и $F''(x)$, причем F' и F'' не меняют знак на отрезке $[a, b]$

$$F'(x) \cdot F''(x) \neq 0, \quad x \in [a, b] \quad (5.40)$$

Зададимся начальным приближением $x_0 \in [a, b]$, причем будем считать, что оно выбрано таким образом, что выполняется условие

$$F(x_0) \cdot F''(x_0) > 0 \quad (5.41)$$

Совокупность условий (5.40) и (5.41) называется **условиями Фурье**. Условие (5.40) фактически означает, что $F'(x)$ и $F''(x)$ будут знакопостоянными функциями.

Будем искать корень (5.1) в виде

$$x_* = x_0 + h \quad (5.42)$$

Воспользуемся разложением в ряд Тейлора функции $F(x)$ в окрестности точки x_0

$$F(x_*) = F(x_0 + h) = F(x_0) + F'(x_0) \cdot h + \frac{1}{2} F''(\zeta) \cdot h^2 \quad (5.43)$$

Будем считать, что величина x_0 достаточно близка к корню x_* и, следовательно, h достаточно мало. Тогда, пренебрегая членом порядка h^2 в разложении и учитывая, что $F(x_*) = 0$, получим

$$F(x_0) + F'(x_0) \cdot h = 0,$$

откуда можно найти

$$h = -F(x_0)/F'(x_0). \quad (5.44)$$

Так как в (5.43) отброшен член, содержащий h^2 , то (5.42) дает не величину x_* , а уточненное значение корня x_1

$$x_1 = x_0 - F(x_0)/F'(x_0).$$

Продолжая этот процесс, получаем следующую итерационную формулу метода Ньютона

$$x_{k+1} = x_k - F(x_k)/F'(x_k) \quad (k = 1, 2, 3, \dots) \quad (5.45)$$

Метод Ньютона имеет простую геометрическую интерпретацию: полученное по формуле (5.45) значение есть точка пересечения касательной

к кривой $y = F(x)$ в точке x_k с осью абсцисс. Действительно, уравнение этой касательной имеет вид

$$y - F(x_k) = F'(x_k)(x - x_k),$$

откуда при $y = 0$ находим пересечение с осью абсцисс $x = x_k - F(x_k) / F'(x_k)$, что в точности совпадает с x_{k+1} , получаемым в методе Ньютона по (5.45).

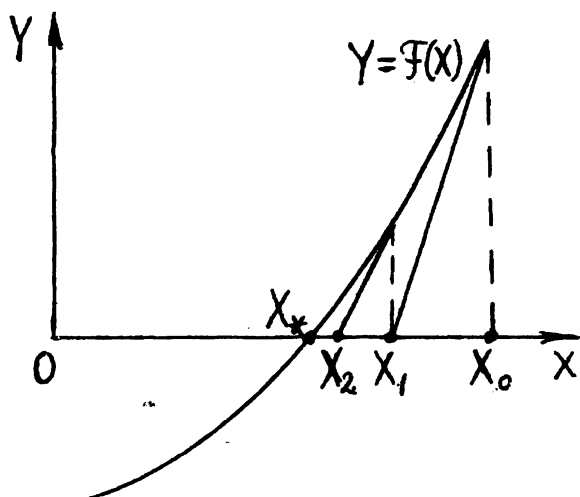


Рис. 5.10

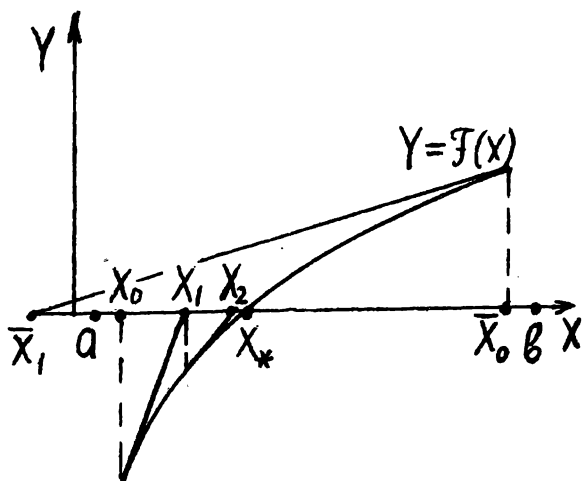


Рис. 5.11

Поэтому метод Ньютона называют иногда методом касательных. На рис. 5.10 и 5.11 приведена геометрическая иллюстрация метода Ньютона.

Следует отметить, что выбор x_0 на рис. 5.10 и 5.11 не является случайным, так как x_0 должно удовлетворять условию (5.41)

$$F(x_0) \cdot F''(x_0) > 0.$$

В противном случае метод может расходиться. Например, если за начальное приближение на рис. 5.11 взять $\bar{x}_0 \approx \bar{b}$, то первое же приближение $\bar{x}_1$ по методу Ньютона выхолит за пределы интервала $[a, b]$, где функция $F(x)$ может быть не определена, или $\bar{x}_1$ не входит в область притяжения корня x . и, следовательно, метод не будет сходиться к этому корню.

5.5.2. Условия сходимости и порядок сходимости

Геометрически очевидно, что выполнение условий Фурье (5.40) и (5.41) для функции $F(x)$ приводит к сходимости метода Ньютона. Приведем без доказательства достаточное условие сходимости метода Ньютона.

Теорема. Пусть $F(x)$ определена и непрерывна на интервале $[a, b]$ и удовлетворяет в нем следующим условиям: 1) $F(a)F(b) < 0$; 2) $F'(x)$ и $F''(x)$ существуют и сохраняют определенные знаки при $x \in [a, b]$. Тогда для любого начального приближения $x_0 \in [a, b]$, удовлетворяющего условию $F(x_0) \cdot F''(x_0) > 0$, приближения x_k , получаемые по формуле Ньютона (5.45), сходятся к единственному значению корня x функции $F(x)$ на отрезке $[a, b]$.

Таким образом, наличие разных знаков у функции $F(x)$ на этом интервале и выполнение для нее условий Фурье достаточно для существования корня $F(x)$ на этом интервале, его единственности и сходимости метода Ньютона с начального значения x_0 , удовлетворяющего (5.41).

Условия приведенной теоремы накладывают серьезное ограничение на поведение функции $F(x)$, поэтому желательно иметь и другие условия сходимости. Одно из них основано на том, что метод Ньютона (5.45) можно рассматривать как метод простой итерации для уравнения $x = \varphi(x)$, где

$$\varphi(x) = x - F(x)/F'(x), \quad (5.46)$$

так как именно $\varphi(x)$ стоит в правой части формулы Ньютона (5.45). Как было показано в разделе 5.4.1, условие сходимости метода простой итерации имеет вид $|\varphi'(x)| < 1$, ($x \in [a, b]$). В нашем случае

$$\varphi'(x) = (x - F(x)/F'(x))' = 1 - \frac{[F'(x)]^2 - F(x)F''(x)}{[F'(x)]^2} = \frac{F(x)F''(x)}{[F'(x)]^2}.$$

Отсюда следует, что условия сходимости метода Ньютона имеют вид

$$\left| \frac{F(x)F''(x)}{[F'(x)]^2} \right| < 1 \quad (5.47)$$

Можно показать, что если x_0 лежит в некоторой δ -окрестности корня $x_0 \in [x_* - \delta, x_* + \delta]$ и для всех x из этой окрестности выполняется условие (5.47), то метод Ньютона сходится к x_* .

Главным достоинством метода Ньютона является то, что при "хорошем" начальном приближении метод сходится очень быстро. В отличие от метода простой итерации, который имеет сходимость первого порядка, **метод Ньютона имеет сходимость второго порядка.*** Можно показать, что последовательные приближения $x_1, x_2, \dots, x_k, \dots$, вычисляемые по методу Ньютона, удовлетворяют условиям

$$|x_{k+1} - x_*| \leq \frac{M_2}{2m_1} |x_k - x_*|^2 \quad (5.48)$$

при $x \in [a, b]$.

Вторая степень в правой части показывает, что метод Ньютона имеет сходимость второго порядка. В (5.48) использованы следующие обозначения: $M_2 = \max |F''(x)|$ при $x \in [a, b]$ (максимальное значение второй производной функции $F(x)$ на отрезке $[a, b]$), $m_1 = \min |F'(x)|$ при $x \in [a, b]$ (минимальное значение первой производной на отрезке $[a, b]$). При выводе (5.48) использовано предположение, что все $x_k \in [a, b]$. Это означает, что начальное приближение x_0 является достаточно хорошим.

Поскольку в (5.48) слева стоит погрешность определения корня на $(k+1)$ -ой итерации, а справа – на k -ой, неравенство (5.48) фактически показывает, что в методе Ньютона количество верных значащих цифр на каждом итерационном шаге, вообще говоря, удваивается, если начальное приближение достаточно хорошее и $M_2 / 2m_1 \leq 1$. Для иллюстрации приведем сравнение метода простой итерации и Ньютона на примере уравнения (5.17) $x^3 + x - 0,5 = 0$, которое было решено в 5.4.4 методом простой итерации (таблица 5.3). Результаты метода Ньютона приведены в таблице 5.4 для того же начального приближения $x_0 = 0,5$.

Таблица 5.4

k	0	1	2	3
x_k	0,5	0,42857143	0,42387218	0,42385380

В таблице верные значащие цифры в каждом приближении подчеркнуты снизу. Сравнение результатов таблиц 5.3 и 5.4 показывает, что метод Ньютона сходится значительно быстрее.

* Порядок сходимости рассмотрен в 5.1.

Недостатком метода Ньютона является зависимость сходимости от хорошего выбора начального приближения x_0 и вида функции $F(x)$. В случае, когда нарушаются условия теоремы о сходимости, последовательность приближений может вести себя непредсказуемым образом. Возможны не только расходимость, но и "зацикливание" метода. Такая "капризность" метода является платой за его высокую точность. Тут уместно привести высказывание специалиста в области вычислительной математики Р.В.Хемминга: "Чем тоньше метод и чем лучше он кажется, тем хуже он может повести себя в случае осложнений с функцией." Проиллюстрируем возможные осложнения при применении метода Ньютона на примере.

5.5.3. Пример "капризного" поведения метода Ньютона

Пример 20. Найти методом Ньютона корни уравнения

$$x^3 - x = 0 \quad (5.49)$$

Очевидно, что (5.49) имеет три корня: $x_1 = -1$, $x_2 = 0$, $x_3 = 1$ (рис.5.12). Выясним, как влияет задание начального приближения на сходимость и к каким корням сходится метод Ньютона при различных x_0 .

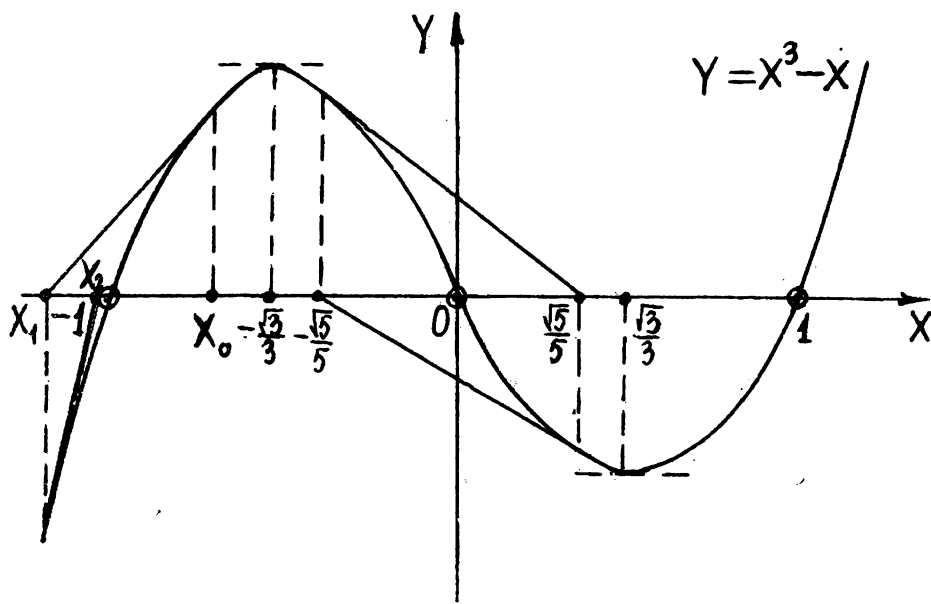


Рис. 5.12

В формуле метода Ньютона (5.45) в знаменателе стоит $F'(x_k)$, поэтому должно быть $F'(x_k) \neq 0$. В нашем случае $F(x) = x^3 - x$, $F'(x) = 3x^2 - 1$,

откуда следует, что $3x^2 - 1 \neq 0$. Значит метод нельзя использовать при $x_0 = \pm \sqrt{3}/3$, так как в этой точке $3x^2 - 1 = 0$. Рассмотрим случаи, возникающие при различных x_0 при $x_0 \neq \pm \sqrt{3}/3$.

1. Рассмотрим значения x_0 на отрезке $-\infty < x_0 < -\sqrt{3}/3$. В этом случае при $x_0 > -1$ условие (5.41)

$$F(x_0) \cdot F''(x_0) > 0 \quad (5.50)$$

не выполняется, так как при $x_0 > -1$ $F(x_0) > 0$, а $F''(x_0) < 0$ (геометрически это видно на рис. 5.12). Однако из геометрического смысла метода Ньютона на рис. 5.12 видно, что после первого шага выполнится условие $x_1 < -1$, а в любой точке, лежащей левее корня $x_1 = -1$, $F(x_1) < 0$, $F''(x_1) < 0$ и поэтому условие (5.50) выполняется $F(x_1) \cdot F''(x_1) > 0$. Так как остальные условия теоремы о сходимости здесь также выполняются, то метод сходится к корню x_1 и сходимость будет монотонной. Аналогично обстоит дело на отрезке $\sqrt{3}/3 < x_0 < +\infty$, так как функция $F(x)$ является нечетной. Разница только в том, что теперь метод Ньютона сходится к корню $x_3 = 1$.

2. Сложнее обстоит дело на отрезке $(-\sqrt{3}/3, +\sqrt{3}/3)$, так как на нем не выполнены условия теоремы о сходимости ($F''(x)$ меняет знак на этом отрезке в точке $x = 0$). Возьмем $x_0 = \sqrt{5}/5$ (или $x_0 = -\sqrt{5}/5$). В этом случае возникает ситуация, когда метод Ньютона "зацикливается": все приближения с нечетными номерами будут равны $-x_0$ ($x_1 = x_3 = x_5 = \dots = x_{2k+1} = -x_0$), а с четными номерами $+x_0$ ($x_2 = x_4 = \dots = x_{2k} = x_0$) (рис. 5.12).

3. Рассмотрим величину x_0 , удовлетворяющую неравенствам $-\sqrt{5}/5 < x_0 < \sqrt{5}/5$. Из геометрического смысла метода Ньютона ясно, что метод будет сходиться к корню $x_2 = 0$. Можно показать, что эта сходимость колеблющаяся: значения x_k будут поочередно то положительными, то отрицательными, причем процесс будет сходиться очень быстро, хотя достаточное условие сходимости здесь не выполняется ($F''(0) = 0$).

4. Рассмотрим теперь значения x_0 , лежащие в одном из двух отрезков: $-\sqrt{3}/3 < x_0 < -\sqrt{5}/5$, $\sqrt{5}/5 < x_0 < \sqrt{3}/3$. Здесь ситуация оказывается наиболее сложной. Из геометрических соображений ясно, что в зависимости от x_0 метод может сходиться либо к корню $x_3 = 1$, либо к корню $x_1 = -1$, но может попасть и в точки $x_k = \pm \sqrt{3}/3$, в которых производная обращается в нуль.

Проанализируем полученные результаты для уравнения (5.49). В случаях 1 и 3 корень, к которому сходится итерационный процесс, лежит в том же интервале, в котором задается начальное приближение x_0 , и такая ситуация представляется довольно естественной. Иначе ведет себя метод Ньютона в случае 4. При некоторых x_0 , лежащих на отрезке $-\sqrt{3}/3 < x_0 < -\sqrt{5}/5$, процесс может сходиться как к "близкому" корню $x_1 = -1$, так и к "далекому"

корню $x_3 = 1$ (при других x_0 из этого же отрезка), "перепрыгивая" через корень $x_2 = 0$.

5.6. Метод секущих

В методе Ньютона требуется вычисление $F(x_k)$ и $F'(x_k)$. Но иногда производная $F'(x)$ вычисляется довольно сложно и желательно ограничиться только вычислением функции $F(x)$. Существуют методы, которые позволяют по двум приближениям x_k и x_{k-1} построить следующее приближение x_{k+1} . Идея заключается в том, чтобы через две точки на кривой $F(x)$ ($x_k, F(x_k)$) и ($x_{k-1}, F(x_{k-1})$) провести прямую и точку пересечения этой прямой с осью x считать следующим приближением. Обозначим $y_k = F(x_k)$ и $y_{k-1} = F(x_{k-1})$, запишем уравнение прямой, проходящей через две точки с координатами (x_k, y_k) и (x_{k-1}, y_{k-1})

$$\frac{y - y_{k-1}}{y_k - y_{k-1}} = \frac{x - x_{k-1}}{x_k - x_{k-1}}$$

Найдем пересечение этой прямой с осью x , положив $y = 0$. После простых преобразований найдем абсциссу пересечения этой прямой с осью x

$$x_{k+1} = x_{k-1} - y_{k-1} \frac{(x_k - x_{k-1})}{(y_k - y_{k-1})} \quad (5.51)$$

*представим себе
прямую
и найдем
ее пересечение с осью*

Применение метода (5.51) зависит от того, как выбраны точки x_{k-1} и x_k . Если начальные точки x_0, x_1 выбраны по разные стороны от корня x_* , например, $x_0 = a, x_1 = b$, то мы получаем всегда сходящийся метод хорд, рассмотренный в 5.3 (рис. 5.13). Если же x_0 и x_1 берутся по одну сторону от корня x_* , то мы получаем метод, называемый методом секущих (рис. 5.14). Достаточные условия сходимости этого метода такие же, как у метода Ньютона: требуется выполнение условий Фурье (5.40), (5.41). В этом случае сходимость метода будет монотонной. Если условия Фурье не выполняются, то метод может расходиться. Показано, что для метода секущих

$$|x_{k+1} - x_*| \leq A |x_k - x_*|^{1.618}$$

*скорость
приближения
к корню*

и, следовательно, его порядок сходимости $n = 1.618$. В отличие от метода Ньютона, в нем не требуется вычисления производных, однако для начала работы метода требуется знать две точки x_0 и x_1 .

5.7. Сравнительная оценка методов

Все рассмотренные методы можно условно разделить на две группы:

1) методы, сходимость которых гарантирована в том случае, если на исходном отрезке $[a, b]$ имеется корень, и никаких специальных условий на вид функции $F(x)$, кроме ее непрерывности, не накладывается;

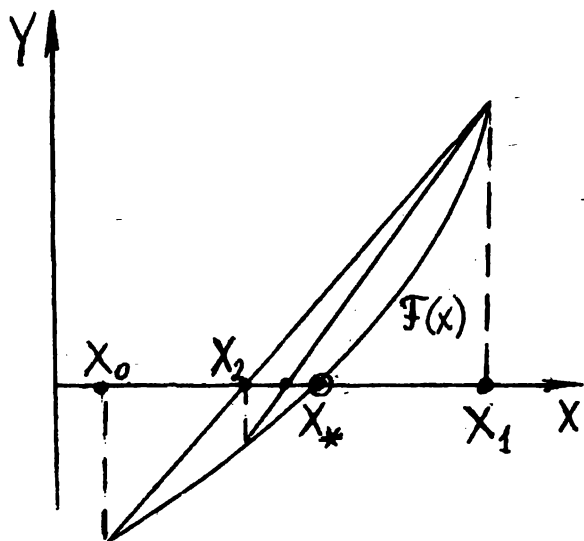


Рис. 5.13. Метод хорд

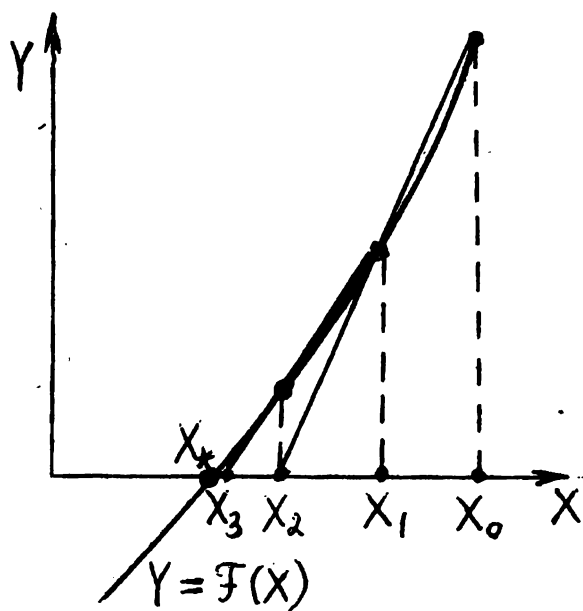


Рис. 5.14. Метод секущих

2) методы, в которых сходимость существует только при некоторых ограничениях на вид функции $F(x)$ и ее производные $F'(x)$, $F''(x)$.

К первой группе из рассмотренных методов относятся метод половинного деления и метод хорд, а ко второй – метод простой итерации, метод Ньютона и метод секущих. Главное отличие этих групп методов проявляется в том, что первые – надежны, но сходятся сравнительно медленно, а вторые зависят от начального приближения и вида функции $F(x)$, но сходятся очень быстро. Поэтому методы второй группы, например, метод Ньютона, удобно употреблять в совокупности с другим методом, сходимость которого (хотя и сравнительно медленная) гарантирована на исходном отрезке $[a, b]$. Таким методом может быть метод половинного деления. В этом случае с помощью нескольких шагов метода половинного деления можно получить интервал $[\alpha, \beta]$ такой, что при $x_0 \in [\alpha, \beta]$ метод Ньютона сходится. После этого 2–3 итерации метода Ньютона позволяют получить значение корня с любой необходимой точностью. Такие гибридные методы используются во многих стандартных программах нахождения корней.

Как отмечалось в 5.2, при нахождении корней желательно получать двусторонние приближения, чтобы на каждом шаге иметь значение корня с избытком и с недостатком, полученные различными методами. Одним из таких удобных способов является совместное применение методов хорд и касательных (Ньютона). Геометрическая иллюстрация совместного использования этих методов показана на рис. 5.15. Строятся две последовательности чисел:

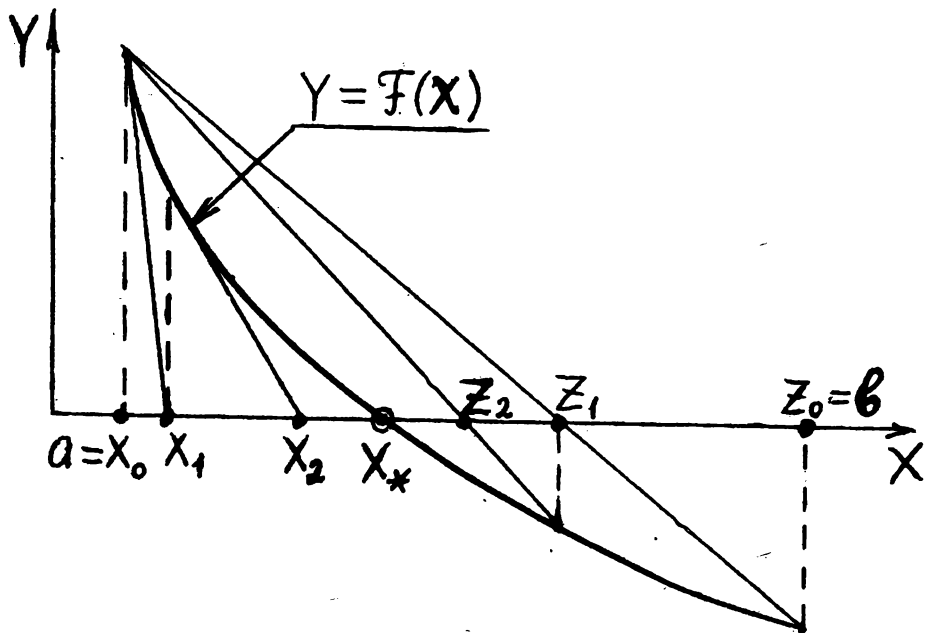


Рис. 5.15

последовательность чисел $x_1, x_2, \dots, x_k$, получаемых в методе Ньютона; последовательность чисел $z_1, z_2, \dots, z_k$, определяемых методом хорд. Очевидно, что если для $F(x)$ выполняются условия Фурье, то обе последовательности будут монотонно сходиться к корню x_* , причем z_k будут давать решения с избытком, а x_k – с недостатком. Таким образом, на каждом шаге процесса для корня уравнения существует надежная двусторонняя оценка.

Важным параметром, по которому можно сравнивать различные методы, является общее количество работы, необходимое в каждом методе. Например, в методе половинного деления на каждой итерации требуется только вычисление середины отрезка c , значения $F(c)$ и вычисление границ нового отрезка. В методе Ньютона помимо значения $F(x)$ требуется вычислять и значение $F'(x)$. Ясно, что если $F(x)$ и $F'(x)$ вычисляются одинаково просто, то можно использовать метод Ньютона. Однако в случае, когда $F(x)$ вычисляется просто, а вычисление $F'(x)$ достаточно громоздко, то имеет смысл использовать метод секущих, в котором не требуется вычислять $F'(x)$ и который сходится так же быстро и при тех же условиях, что и метод Ньютона.

В некоторых случаях очень сложно вычисляется сама функция $F(x)$, для получения которой необходимо выполнение очень большого количества машинных операций. В этом случае желательно получать очень быстро сходящиеся итерационные методы, использующие как можно больше уже полученной информации. Речь идет о том, чтобы, получив, например, результаты $x_1, x_2, \dots, x_k$ нескольких итерационных шагов, использовать их для более точного приближения корня. Кратко изложим один из наиболее известных таких методов, называемый **методом Стеффенсена** (метод, позволяющий значительно ускорить сходимость итерационных методов с помощью специально разработанной схемы расчетов).

Пусть имеется уравнение $x = \varphi(x)$, которое мы решаем методом итераций. Задав x_0 , определим $x_1 = \varphi(x_0)$ и $x_2 = \varphi(x_1)$. Стеффенсен предложил, используя три величины x_0, x_1, x_2 , построить специальным образом новое приближение y_1 , которое будет гораздо ближе к значению корня, чем x_2 . Для этого к уравнению

$$F(x) \equiv x - \varphi(x) = 0 \quad (5.52)$$

применяют метод секущих, взяв за исходные точки x_0 и x_1 . В результате получится новое приближение

$$y_1 = \frac{x_0 F(x_1) - x_1 F(x_0)}{F(x_1) - F(x_0)} \quad (5.53)$$

Так как $F(x_0) = x_0 - \varphi(x_0) = x_0 - x_1$, а $F(x_1) = x_1 - \varphi(x_1) = x_1 - x_2$, то (5.53) можно записать в виде

$$y_1 = \frac{x_0 x_2 - x_1^2}{x_0 - 2x_1 + x_2} \quad (5.54)$$

Взяв теперь y_1 в качестве нового начального приближения $\bar{x}_0 = y_1$ и вычислив снова $\bar{x}_1 = \varphi(\bar{x}_0)$, $\bar{x}_2 = \varphi(\bar{x}_1)$, мы можем получить следующее приближение по методу Стеффенсена. Показано, что этот метод является фактически методом итераций $y = \psi(y)$, где $\psi(y)$ имеет следующий вид

где надо положить $y_0 = x_0$, а y_1 определяется по формуле (5.54).

ГЛАВА 6. СИСТЕМЫ НЕЛИНЕЙНЫХ УРАВНЕНИЙ

Многие практические задачи при их математическом моделировании сводятся к решению системы нелинейных уравнений. Система n нелинейных уравнений с n неизвестными в общем случае имеет вид

Решением системы уравнений (6.1) называется n чисел $x_1, x_2, \dots, x_n$, которые при подстановке их в (6.1) обращают все уравнения в нули.

Эта связь проявляется в том, что каждую из этих задач можно свести к решению другой. Пусть, например, вектор $\bar{x}_* \{x_{1*}, x_{2*}, \dots, x_{n*}\}$ является решением системы (6.1). Построим функцию

Так как $\bar{x}_*$ обращает в нуль все уравнения системы (6.1), то $\bar{x}_*$ обращает функцию $\bar{Q}(x_1, x_2, \dots, x_n)$ в нуль. Поскольку $Q \geq 0$, то $\bar{x}_*$ обращает функцию Q в минимум, и этот минимум равен нулю. Справедливо и обратное утверждение. Пусть совокупность чисел $\{x_1, x_2, \dots, x_n\}$ обращает функцию $Q(x_1, x_2, \dots, x_n)$

в минимум, равный нулю. Тогда каждый член суммы, входящий в (6.3), обращается в нуль и, следовательно, совокупность $x_1, x_2, \dots, x_n$ является решением системы (6.1). Таким образом, задача (6.1) сводится к нахождению минимума (6.3).

В свою очередь, для задачи минимизации (6.2) точка минимума является решением системы нелинейных уравнений

$$\begin{cases} \frac{\partial u(x_1, x_2, \dots, x_n)}{\partial x_1} = 0 \\ \dots\dots\dots \\ \frac{\partial u(x_1, x_2, \dots, x_n)}{\partial x_n} = 0 \end{cases}$$

Возможность сведения одной задачи к другой не означает, что можно ограничиться рассмотрением только одной из них. Каждая из них является весьма сложной и имеет многочисленные подводные камни, которые проявляются при практической реализации существующих методов их решения. По-видимому, не существует никаких универсальных методов решения этих задач. Установление связи между задачами (6.1) и (6.2) удобно тем, что в каждом конкретном случае мы можем использовать ту из них, которая быстрее приводит к цели при располагаемых ресурсах вычислительной техники и математического обеспечения.

Приведем пример совместного использования задач (6.1) и (6.3). Пусть нам необходимо решить систему нелинейных уравнений (6.1). Методы, используемые для ее решения, как правило, требуют знания хорошего начального приближения для сходимости существующих итерационных методов. Это начальное приближение может быть получено с помощью нескольких шагов какого-либо метода минимизации функции вида (6.3). Удобство такого подхода заключается в том, что методы, применяемые при минимизации функции многих переменных, обычно имеют более широкую область сходимости, чем методы решения нелинейных уравнений. В то же время последние (методы решения нелинейных уравнений) обычно обладают лучшей скоростью сходимости при наличии хорошего начального приближения. Поэтому, получив хорошее начальное приближение методом минимизации, мы далее уточняем корни системы уравнений одним из эффективных методов (например, методом Ньютона).

В предыдущей главе в разделе 5.1 были отмечены характерные особенности нелинейных задач на примере одного нелинейного уравнения. В еще большей степени трудности решения этих задач проявляются при решении систем нелинейных уравнений. Это проявляется в следующем:

- 1) определение количества корней системы и их отделение проводить здесь гораздо сложнее в силу невозможности при больших n использовать простые геометрические соображения;
- 2) многие надежные методы нахождения корня одного нелинейного уравнения, имеющие гарантированную сходимость, такие как метод

(6.5)

Итерационный процесс продолжается до тех пор, пока изменение всех неизвестных на двух последовательных итерациях не станут малыми

(6.6)

Уравнение (6.4) удобно записать в векторном виде

(6.7)

где $\bar{x}$ – вектор-столбец искомого решения, а Φ – вектор-функция

$$\bar{x} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} \quad \Phi = \begin{pmatrix} \varphi_1 \\ \varphi_2 \\ \dots \\ \varphi_n \end{pmatrix}$$

Итерационный процесс (6.5) можно также записать в векторном виде

(6.8)

Условия сходимости итерационного метода (6.5) (или в векторном виде (6.8)) определяется принципом сжатых отображений: если отображение (6.7), задаваемое вектор-функцией $\Phi(\bar{x})$ будет сжатым, то итерационный процесс (6.8) сходится при условии, что начальное приближение $\bar{x}^{(0)}$ достаточно близко к корню.

Приведем одно из достаточных условий сходимости метода простой итерации(6.5). Рассмотрим матрицу Якоби для преобразования (6.4)

$$J(x_1, \dots, x_n) = \begin{bmatrix} \frac{\partial \varphi_1}{\partial x_1} & \frac{\partial \varphi_1}{\partial x_2} & \dots & \frac{\partial \varphi_1}{\partial x_n} \\ \frac{\partial \varphi_2}{\partial x_1} & \frac{\partial \varphi_2}{\partial x_2} & \dots & \frac{\partial \varphi_2}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial \varphi_n}{\partial x_1} & \frac{\partial \varphi_n}{\partial x_2} & \dots & \frac{\partial \varphi_n}{\partial x_n} \end{bmatrix} \quad (6.9)$$

Дифференциал функции

Матрица Якоби (или якобиан) зависит от переменных $x_1, x_2, \dots, x_n$. Можно показать, что если матрица Якоби в точке $\bar{x}_*$, которая дает решение системы (6.4), и в некоторой ее окрестности имеет все собственные значения, по модулю меньшие единицы $|\lambda_i| < 1$ ($i = 1, 2, \dots, n$), то метод простой итерации сходится. Для выполнения этого условия достаточно, например, выполнения следующих неравенств для элементов матрицы Якоби

$$\begin{cases} \left| \frac{\partial \varphi_1}{\partial x_1} \right| + \left| \frac{\partial \varphi_1}{\partial x_2} \right| + \dots + \left| \frac{\partial \varphi_1}{\partial x_n} \right| \leq q \\ \left| \frac{\partial \varphi_2}{\partial x_1} \right| + \left| \frac{\partial \varphi_2}{\partial x_2} \right| + \dots + \left| \frac{\partial \varphi_2}{\partial x_n} \right| \leq q \\ \dots \dots \dots \\ \left| \frac{\partial \varphi_n}{\partial x_1} \right| + \left| \frac{\partial \varphi_n}{\partial x_2} \right| + \dots + \left| \frac{\partial \varphi_n}{\partial x_n} \right| \leq q \end{cases}, \quad (6.10)$$

где q – некоторое положительное число, меньшее единицы, т.е. $0 < q < 1$. Неравенства (6.10) должны выполняться в точке $\bar{x}_*$ и в некоторой ее окрестности.

Возможно естественное обобщение метода простой итерации (6.5), аналогичное методу Зейделя для систем линейных уравнений. Оно заключается в том, что переменные x_i , уже полученные на данном итерационном шаге, используются для получения других величин на этом шаге. Соответствующие формулы имеют вид

$$\begin{cases} x_1^{(k+1)} = \varphi_1(x_1^{(k)}, x_2^{(k)}, \dots, x_{n-1}^{(k)}, x_n^{(k)}) \\ x_2^{(k+1)} = \varphi_2(x_1^{(k+1)}, x_2^{(k)}, \dots, x_{n-1}^{(k)}, x_n^{(k)}) \\ \dots \dots \dots \\ x_n^{(k+1)} = \varphi_n(x_1^{(k+1)}, x_2^{(k+1)}, \dots, x_{n-1}^{(k+1)}, x_n^{(k)}) \end{cases} \quad (6.11)$$

Если оба метода (6.5) и (6.11) сходятся, то метод (6.11) сходится, вообще говоря, быстрее, так как получаемые на каждом шаге итерации новые значения переменных сразу используются в дальнейших вычислениях на этом шаге. Приведем пример использования рассмотренных методов.

Пример 21. Решить методом итераций систему двух нелинейных уравнений с двумя неизвестными

$$\begin{cases} y \cdot (x - 1) = 1 \\ x^2 = y^2 + 1 \end{cases} \quad (6.12)$$

Решение. Оба уравнения (6.12) представляют собой гиперболы, вид которых приведен на рис. 6.1. Из рисунка видно, что существует два вещественных корня. Один из них расположен в первой четверти ($x_1 > 0$, $y_1 > 0$), а другой – в третьей ($x_2 < 0$, $y_2 < 0$).

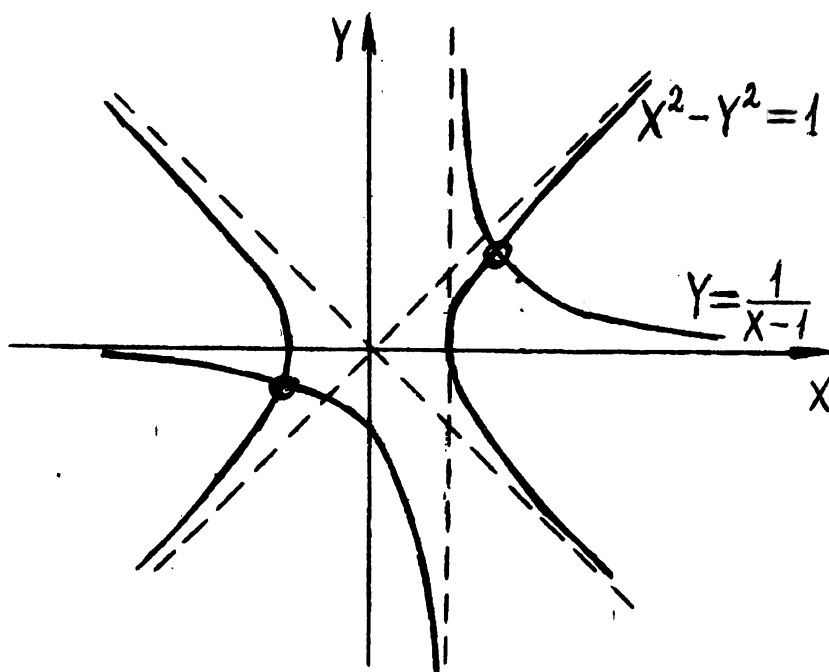


Рис. 6.1

Для нахождения первого корня запишем систему (6.12) в виде, удобном для итераций. Для этого выразим x из первого уравнения, а y – из второго.

$$\begin{cases} x = 1 + \frac{1}{y} \\ y = \sqrt{x^2 - 1} \end{cases} \quad (6.13)$$

Для численного нахождения корня (x_1, y_1) применим итерационный процесс (6.11):

$$\begin{cases} x^{(k+1)} = 1 + \frac{1}{y^{(k)}} \\ y^{(k+1)} = \sqrt{(x^{(k+1)})^2 - 1} \end{cases}$$

Из рис. 6.1 видно, что $x_1 > 0, y_1 > 0$. Возьмем $x^{(0)} = 2, y^{(0)} = \sqrt{(x^{(0)})^2 - 1} = 1,7321$. Результаты, получаемые с помощью итерационного метода, приведены в таблице 6.1.

Таблица 6.1

k	0	1	2	3	5	...	18
$x^{(k)}$	2,0	1,5773	1,8198	1,6577	1,6926	...	1,7167
$y^{(k)}$	1,7321	1,2198	1,5204	1,3221	1,3656	...	1,3954

Видно, что итерационный процесс (6.13) сходится. Сравнивая эти результаты с приведенными ниже результатами метода Ньютона для решения этой же задачи, видно, что метод простой итерации сходится довольно медленно. Примерно одинаковых по точности результатов можно получить за 18 шагов метода простой итерации и всего за 3 шага метода Ньютона.

6.3. Метод Ньютона и его модификации

Метод Ньютона – метод, который сводит решение системы нелинейных уравнений к последовательному решению систем линейных уравнений для нахождения поправок к предыдущим приближениям.

Рассмотрим общий вид нелинейной системы (6.1). Зададим начальное приближение $\bar{x}^{(0)} \{x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}\}$. Будем искать решение системы (6.1) в виде

$$x_{1*} = x_1^{(0)} + \Delta x_1^{(0)}, \quad x_{2*} = x_2^{(0)} + \Delta x_2^{(0)}, \quad x_{n*} = x_n^{(0)} + \Delta x_n^{(0)}. \quad (6.14)$$

Разложим левые части системы (6.1) в ряд Тейлора в точке $\bar{x}^{(0)}$, считая величины $\Delta x_i^{(0)}$ малыми и в силу этого ограничиваясь только линейными членами в ряде Тейлора. Для функции $f_i(x_1, x_2, \dots, x_n)$ получим

производные удовлетворяли некоторым ограничениям. Подробные сведения о сходимости метода Ньютона приведены в монографии [7].

Несмотря на быструю сходимость, метод Ньютона имеет существенный недостаток: на каждом его шаге требуется вычислять n^2 частных производных

$\frac{\partial f_i}{\partial x_j}$, определяющих коэффициенты при неизвестных в линейной системе (6.20) (якобиан). В то же время в методе простой итерации (6.5) требуется вычислять всего n функций $\varphi_i(x_1, x_2, \dots, x_n)$ ($i = 1, 2, \dots, n$). Чтобы компенсировать этот недостаток метода Ньютона, были предложены различные его упрощения. Идея большинства из этих упрощений заключается в том, чтобы матрицу коэффициентов линейной системы (6.20) вычислять не на каждом итерационном шаге, а лишь на некоторых из них, считая на остальных шагах все элементы якобиана постоянными, равными их последним вычисленным значениям на том шаге, на котором производился пересчет всех членов. В простейшем случае якобиан вычисляется один раз, в начальной точке $\bar{x}^{(0)}$, а затем якобиан остается постоянным в течение всего итерационного процесса. Такой метод обычно называют упрощенным методом Ньютона. Матрица коэффициентов линейной системы (6.20) в этом случае остается постоянной, и в процессе итераций изменяются только правые части системы, которые пересчитываются на каждом шаге итераций и решения системы $\Delta x_i^{(k)}$, после нахождения которых происходит пересчет $x_i^{(k+1)}$ по формулам (6.21). Конечно, этот метод сходится медленнее, чем метод Ньютона, но на каждом итерационном шаге требует гораздо меньше арифметических операций.

Рассмотрим пример применения метода Ньютона.

Пример 22. Решить методом Ньютона систему

$$\begin{cases} f_1 = x_2(x_1 - 1) - 1 = 0 \\ f_2 = x_1^2 - x_2^2 - 1 = 0 \end{cases} \quad (6.22)$$

Решение. Система (6.22) была решена в примере 21 методом итерации. Геометрическое изображение системы приведено на рис. 6.1. Система имеет два корня: в первой четверти ($x_1 > 0, x_2 > 0$) и в третьей четверти ($x_1 < 0, x_2 < 0$). Найдём методом Ньютона положительный корень. В качестве начального приближения примем $x_1^{(0)} = 2, x_2^{(0)} = 1$. Вычислим частные производные,

входящие в якобиан $\frac{\partial f_1}{\partial x_1} = x_2; \frac{\partial f_1}{\partial x_2} = x_1 - 1; \frac{\partial f_2}{\partial x_1} = 2x_1; \frac{\partial f_2}{\partial x_2} = -2x_2$. В данном конкретном случае система (6.20) имеет вид

$$\begin{cases} (x_2)_k \Delta x_1^{(k)} + (x_1 - 1)_k \Delta x_2^{(k)} = 1 - x_2^{(k)}(x_1^{(k)} - 1) \\ (2x_1)_k \Delta x_1^{(k)} + (-2x_2)_k \Delta x_2^{(k)} = 1 - (x_1^{(k)})^2 + (x_2^{(k)})^2 \end{cases} \quad (6.23)$$

Подставляя в коэффициенты системы и правые части начальное приближение $x_1^{(0)} = 2$, $x_2^{(0)} = 1$, получим систему линейных уравнений

$$\begin{cases} 1 \cdot \Delta x_1^{(0)} + 1 \cdot \Delta x_2^{(0)} = 0 \\ 4 \cdot \Delta x_1^{(0)} - 2 \cdot \Delta x_2^{(0)} = 2 \end{cases}$$

Решим систему методом Гаусса. Для этого из второго уравнения системы вычтем первое, умноженное на 4. Получим систему с верхней треугольной матрицей

$$\begin{cases} \Delta x_1^{(0)} + \Delta x_2^{(0)} = 0 \\ -6\Delta x_2^{(0)} = -2 \end{cases}$$

откуда найдем $\Delta x_2^{(0)} = 2/6 = 0,3333$, $\Delta x_1^{(0)} = -0,3333$. Используя (6.21), находим уточненные значения неизвестных

$$\begin{cases} x_1^{(1)} = x_1^{(0)} + \Delta x_1^{(0)} = 2 - 0,3333 = 1,6667 \\ x_2^{(1)} = x_2^{(0)} + \Delta x_2^{(0)} = 1 + 0,3333 = 1,3333 \end{cases}$$

На этом первый шаг метода Ньютона заканчивается. Подставляя полученные значения $x_1^{(1)}$ и $x_2^{(1)}$ в коэффициенты и правые части системы (6.23), получим систему линейных уравнений для определения величин $\Delta x_1^{(1)}$ и $\Delta x_2^{(1)}$

$$\begin{cases} 1,3333\Delta x_1^{(1)} + 0,6667\Delta x_2^{(1)} = 0,1111 \\ 3,3334\Delta x_1^{(1)} - 2,6666\Delta x_2^{(1)} = 0,0000 \end{cases}$$

Из второго уравнения вычтем первое, умноженное на величину $3,3334/1,3333 = 2,5000$. Получим систему в треугольном виде

$$\begin{cases} 1,3333\Delta x_1^{(1)} + 0,6667\Delta x_2^{(1)} = 0,1111 \\ -4,3334\Delta x_2^{(1)} = -0,2778 \end{cases}$$

Из второго уравнения определим $\Delta x_2^{(1)} = 0,0641$. Подставив это значение

в первое уравнение, найдем $\Delta x_1^{(1)}$

$$1,3333 \Delta x_1^{(1)} = 0,1111 - 0,6667 - 0,0641 = 0,0684, \Delta x_1^{(1)} = 0,0513$$

Найдем теперь уточненные значения $x_1^{(2)}$ и $x_2^{(2)}$ по формулам (6.21)

$$\begin{cases} x_1^{(2)} = x_1^{(1)} + \Delta x_1^{(1)} = 1,6667 + 0,0513 = 1,718 \\ x_2^{(2)} = x_2^{(1)} + \Delta x_2^{(1)} = 1,3333 + 0,0641 = 1,3974 \end{cases}$$

Представим результаты в виде таблицы 6.2, причем для сравнения приведем значения решения после трех шагов Ньютона.

Таблица 6.2

k	0	1	2	3	...	6	
$x_1^{(k)}$	2	1,6667	1,718	1,7167			классический метод Ньютона
$x_2^{(k)}$	1	1,3333	1,3974	1,3953			
$ f_1 $	0	0,1111	0,0033	0,0001			
$ f_2 $	2	0,0002	0,0012	0,0002			
$x_1^{(k)}$	2	1,6667	1,7037	...	...	1,7166	упрощенный метод Ньютона
$x_2^{(k)}$	1	1,3333	1,4078	...	...	1,3954	
$ f_1 $	0	0,1111	0,0093	...	...	0,00006	
$ f_2 $	2	00002	0,0793	...	...	0,00004	

Решим теперь эту же задачу упрощенным методом Ньютона. Для этого матрицу коэффициентов в (6.23) надо вычислить один раз при $x_1^{(0)}$ и $x_2^{(0)}$. Получим систему уравнений (6.23) в упрощенном виде

$$\begin{cases} 1 \cdot \Delta x_1^{(k)} + 1 \cdot \Delta x_2^{(k)} = 1 - x_2^{(k)}(x_1^{(k)} - 1) \\ 4 \cdot \Delta x_1^{(k)} - 2 \cdot \Delta x_2^{(k)} = 1 - (x_1^{(k)})^2 + (x_2^{(k)})^2 \end{cases} \quad (6.24)$$

Первый шаг упрощенного метода Ньютона будет полностью совпадать с первым шагом обычного метода Ньютона, расчеты для которого проведены выше. Получим

$$x_1^{(1)} = 1,6667, \quad x_2^{(1)} = 1,3333 \quad (6.25)$$

Сделаем теперь второй шаг упрощенного метода. Для этого подставим полученные значения (6.25) в правые части системы (6.24). Система (6.24) примет вид.

$$\begin{cases} 1 \cdot \Delta x_1^{(1)} + 1 \cdot \Delta x_2^{(1)} = 1 - 1,3333(1,6667 - 1) = 0,1111 \\ 4 \cdot \Delta x_1^{(1)} - 2 \cdot \Delta x_2^{(1)} = 0 \end{cases}$$

Вычитая из второго уравнения первое, умноженное на 4, получим систему в треугольном виде

$$\begin{cases} \Delta x_1^{(1)} + \Delta x_2^{(1)} = 0,1111 \\ - 6 \cdot \Delta x_2^{(1)} = -0,4444, \end{cases}$$

откуда легко найти, что $\Delta x_2^{(1)} = 0,07407$, $\Delta x_1^{(1)} = 0,0370$. Из (6.21) находим уточненное решение

$$\begin{cases} x_1^{(2)} = x_1^{(1)} + \Delta x_1^{(1)} = 1,6667 + 0,03703 = 1,7037 \\ x_2^{(2)} = x_2^{(1)} + \Delta x_2^{(1)} = 1,3337 + 0,07407 = 1,4078 \end{cases}$$

Без выкладок приведем результаты следующего шага упрощенного метода Ньютона: $x_1^{(3)} = 1,7199$, $x_2^{(3)} = 1,4008$. Результаты приведены в таблице 6.2, причем для сравнения приведем значения решения упрощенным методом Ньютона на шестом шаге. Видно, что упрощенный метод в данном случае сходится примерно в два раза медленнее, но зато не требует на каждом шаге решения линейной системы (6.23). Для этого достаточно один раз найти обратную матрицу A^{-1} для системы (6.23) и находить решение (6.23) по формуле

$$\bar{x}^{(k)} = A^{-1} \cdot \bar{b}^{(k)},$$

где $\bar{b}^{(k)}$ – это вектор правых частей системы на k -ом шаге.

ПЕРЕЧЕНЬ УМЕНИЙ

№ п/п	Умение	Алгоритмы
1	Решение систем линейных уравнений $A\bar{x} = \bar{b}$ прямыми методами	1. Свести систему к треугольному виду с помощью эквивалентных преобразований (раздел 2.2). 2. Обратным ходом метода Гаусса найти решение системы (раздел 2.2). 3. Выписать матрицы L и U такие, что $A = LU$ (раздел 2.3). 4. Решить исходную систему с помощью LU – разложения
2	Решение систем линейных уравнений $A\bar{x} = \bar{b}$ итерационными методами	1. Привести систему к виду удобному для итераций, с учетом условий сходимости (раздел 3.1.3). 2. Сделать несколько итераций методом Зейделя (раздел 3.3)
3	Нахождение обратной матрицы численным методом	1. Написать расширенную матрицу, содержащую исходную и единичную матрицы (раздел 4.1). 2. Свести исходную матрицу методом Гаусса к верхней треугольной матрице. 3. Решая треугольную систему с несколькими правыми частями, определить все столбцы обратной матрицы (раздел 4.1). 4. Для контроля проверить выполнение равенства $AA^{-1} = E$
4	Определение максимального по модулю собственного значения и соответствующего собственного вектора степенным методом	1. Задать начальный ненулевой вектор. 2. Нормировать его, сделав длину равной единице (раздел 4.4). 3. Умножить исходную матрицу на нормированный вектор. 4. С полученным вектором проделать снова пп.2 и 3 (несколько раз).
5	Нахождение корней одного нелинейного уравнения $f(x) = 0$	1. Отделить корни (раздел 5.1). 2. Решить методом половинного деления (или хорд), сделав несколько шагов (раздел 5.3). 3. Оценить сходимость метода Ньютона и найти хорошее начальное приближение (раздел 5.5). 4. Получить двустороннюю оценку для корня, сделав 1 или 2 шага комбинированным методом Ньютона и хорд (раздел 5.5, 5.7)
6	Нахождение корней системы нелинейных уравнений	1. Проанализировать систему, найти количество корней и их расположение. 2. Решить систему методом Ньютона (раздел 6.3)

ТРЕНИНГ УМЕНИЙ

1. Пример выполнения упражнения тренинга на умение 1

Задание

а) Решить прямым методом Гаусса систему линейных уравнений:

$$\begin{cases} x_1 - 2x_2 + 3x_3 = 4 \\ 2x_1 + x_2 + x_3 = 3 \\ 5x_1 + x_2 + x_3 = 6 \end{cases}$$

б) Найти LU-разложение для матрицы ее коэффициентов.

Решение

1. Сведем систему к треугольному виду с помощью эквивалентных преобразований. Для этого из второй строки вычтем первую, умноженную на 2, а из третьей строки вычтем первую, умноженную на 5 (запомним эти два числа – 2 и 5, так они будут стоять в матрице L : $l_{21} = 2$, $l_{31} = 5$). В результате получится эквивалентная система, у которой во втором и третьем уравнении отсутствует неизвестная x_1 , что означает, что соответствующие коэффициенты матрицы у преобразованной системы равны нулю. Преобразованная система примет вид:

$$\begin{cases} x_1 - 2x_2 + 3x_3 = 4 \\ 5x_2 - 5x_3 = -5 \\ 11x_2 - 14x_3 = -14 \end{cases}$$

Напомним, что все преобразования производятся с полной строкой, включающей член в правой части. Теперь из третьей строки вычтем вторую, умноженную на $11/5$ (опять запомним это число, так как будет $l_{32} = 11/5 = 2,2$). В результате получаем систему с верхней треугольной матрицей:

$$\begin{cases} x_1 - 2x_2 + 3x_3 = 4 \\ 5x_2 - 5x_3 = -5 \\ -3x_3 = -3 \end{cases}$$

На этом заканчивается прямой ход метода Гаусса.

2. Обратный ход заключается в определении неизвестных из полученной треугольной системы. Из третьего уравнения находим $x_3 = 1$. Подставив x_3 во второе уравнение, найдем $x_2 = 0$. Подставив x_3 и x_2 в первое уравнение, определим $x_1 = 4 + 2x_2 - 3x_3 = 1$. Итак, решением системы будет $x_1 = 1, x_2 = 0, x_3 = 1$.

3. Перейдем к нахождению LU-разложения матрицы A. Матрица U – это верхняя треугольная матрица у преобразованной системы

$$U = \begin{bmatrix} 1 & -2 & 3 \\ 0 & 5 & -5 \\ 0 & 0 & -3 \end{bmatrix}$$

Матрица L – это нижняя треугольная матрица, у которой на главной диагонали стоят единицы, а коэффициенты ниже главной диагонали равны тем числам, на которые умножались ведущие строки при преобразовании системы (мы эти числа запоминали: $l_{21} = 2, l_{31} = 5, l_{32} = 2,2$).

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 5 & 2,2 & 1 \end{bmatrix}$$

Убедимся, что действительно $A = LU$.

$$LU = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 5 & 2,2 & 1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 3 \\ 0 & 5 & -5 \\ 0 & 0 & -3 \end{bmatrix} = \begin{bmatrix} 1 & -2 & 3 \\ 2 & 1 & 1 \\ 5 & 1 & 1 \end{bmatrix} = A$$

4. Решим теперь исходную систему с помощью LU-разложения. Систему $A\bar{x} = \bar{b}$ теперь можно записать в виде $LU\bar{x} = \bar{b}$. Введем неизвестный пока вектор $\bar{y} = U\bar{x}$. Тогда система $LU\bar{x} = \bar{b}$ запишется в виде двух линейных систем: $L\bar{y} = \bar{b}, U\bar{x} = \bar{y}$. Поскольку обе матрицы L и U треугольные, то обе системы можно решить без труда. При этом сначала решаем первую систему $L\bar{y} = \bar{b}$, откуда находим вектор $\bar{y}$, а затем из второй системы находим решение системы – вектор $\bar{x}$. Первая система имеет вид:

$$\begin{cases} y_1 & = & 4 \\ 2y_1 + y_2 & = & 3 \\ 5y_1 + y_2 + y_3 & = & 6 \end{cases}$$

Из первого уравнения находим $y_1 = 4$. Подставляя y_1 во второе уравнение, находим $y_2 = 3 - 8 = -5$. Подставляя y_1 и y_2 в третье уравнение, определяем $y_3 = 6 - 20 + 11 = -3$. Теперь, для нахождения $\bar{x}$ решаем вторую систему, правая часть которой равна найденному вектору $\bar{y}$,

$$\begin{cases} x_1 - 2x_2 + 3x_3 = 4 \\ 5x_2 - 5x_3 = -5 \\ -3x_3 = -3 \end{cases}$$

Эта система уже была решена выше в п.1. Ее решение: $x_1 = 1, x_2 = 0, x_3 = 1$.
Ответ: а) $x_1 = 1, x_2 = 0, x_3 = 1$.

$$\text{б) } L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 5 & 2,2 & 1 \end{bmatrix} \text{ и } U = \begin{bmatrix} 1 & -2 & 3 \\ 0 & 5 & -5 \\ 0 & 0 & -3 \end{bmatrix}$$

Решите самостоятельно следующие задания:

Задание 1.1

Дана система линейных уравнений:

$$\begin{cases} x_1 + 5x_2 + x_3 = 11 \\ 4x_1 - x_2 + x_3 = 2 \\ x_1 + x_2 + 6x_3 = 3 \end{cases}$$

- а) Решить систему методом Гаусса;
- б) Найти LU-разложение матрицы системы;
- с) Решить исходную систему с помощью LU-разложения.

✓ Задание 1.2

Дана система линейных уравнений:

$$\begin{cases} x_1 + 5x_2 + x_3 = 12 \\ 4x_1 - x_2 + x_3 = 3 \\ 2x_1 + x_2 + 8x_3 = 12 \end{cases}$$

- а) Решить систему методом Гаусса;
- б) Найти LU-разложение матрицы системы;
- с) Решить исходную систему с помощью LU-разложения.

Задание 1.3

Дана система линейных уравнений:

$$\begin{cases} x_1 + 5x_2 + x_3 = 31 \\ 4x_1 - x_2 + x_3 = 4 \\ 3x_1 + x_2 + 10x_3 = 25 \end{cases}$$

- a) Решить систему методом Гаусса;
- b) Найти LU-разложение матрицы системы;
- c) Решить исходную систему с помощью LU-разложения.

2. Пример выполнения упражнения тренинга на умение 2

Задание

Дана система линейных уравнений:

$$\begin{cases} x_1 - 2x_2 + 3x_3 = 4 \\ 2x_1 + x_2 + x_3 = 3 \\ x_1 + 5x_2 + x_3 = 2 \end{cases}$$

- Привести систему к виду, удобному для итераций, с учетом условий сходимости;
- Сделать 3 шага методом Зейделя, взяв $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$.

Решение

1. Приведем систему к виду, удобному для итераций, с учетом условий сходимости.

Матрица исходной системы не обладает свойством диагонального преобладания, так как $|1| < |-2| + |3|$, $|1| < |2| + |1|$, $|1| < |5| + |1|$.

Однако, если поменять местами уравнения, сделав первое уравнение третьим, второе – первым, а третье – вторым, то мы получим систему:

$$\begin{cases} 2x_1 + x_2 + x_3 = 3 \\ x_1 + 5x_2 + x_3 = 2 \\ x_1 - 2x_2 + 3x_3 = 4 \end{cases}$$

которая обладает этим свойством, так как $|2| \geq |1| + |1|$, $|5| \geq |1| + |1|$, $|3| \geq |1| + |-2|$.

2. Приведем теперь систему к виду, удобному для итераций, выразив из первого уравнения x_1 , из второго – x_2 , а из третьего – x_3 .

$$\begin{cases} x_1 = (3 - x_2 - x_3) / 2 \\ x_2 = (2 - x_1 - x_3) / 5 \\ x_3 = (4 - x_1 + 2x_2) / 3 \end{cases}$$

Первый шаг метода Зейделя вычисляем по формулам:

$$x_1^{(1)} = (3 - x_2^{(0)} - x_3^{(0)}) / 2 = (3 - 0 - 0) / 2 = 1,5$$

$$x_2^{(1)} = (2 - x_1^{(1)} - x_3^{(0)}) / 5 = (2 - 1,5 - 0) / 5 = 0,1$$

$$x_3^{(1)} = (4 - x_1^{(1)} + 2x_2^{(1)}) / 3 = (4 - 1,5 + 2 \cdot 0,1) / 3 = 0,9$$

Вычисление второго приближения:

$$x_1^{(2)} = (3 - x_2^{(1)} - x_3^{(1)}) / 2 = (3 - 0,1 - 0,9) / 2 = 1$$

$$x_2^{(2)} = (2 - x_1^{(2)} - x_3^{(1)}) / 5 = (2 - 1 - 0,9) / 5 = 0,02$$

$$x_3^{(2)} = (-x_1^{(2)} + 2x_2^{(2)}) / 3 = (-1 + 2 \cdot 0,02) / 3 = -0,966667$$

Вычисление третьего приближения:

$$x_1^{(3)} = (3 - x_2^{(2)} - x_3^{(2)}) / 2 = (3 - 0,02 - (-0,966667)) / 2 = 2,473333$$

$$x_2^{(3)} = (2 - x_1^{(3)} - x_3^{(2)}) / 5 = (2 - 2,473333 - (-0,966667)) / 5 = 0,493333$$

$$x_3^{(3)} = (-x_1^{(3)} + 2x_2^{(3)}) / 3 = (-2,473333 + 2 \cdot 0,493333) / 3 = -1,493333$$

Для сравнения укажем, что точное решение $x_1 = 1, x_2 = 0, x_3 = 1$.

Ответ: а)
$$\begin{cases} x_1 = (3 - x_2 - x_3)/2 \\ x_2 = (2 - x_1 - x_3)/5 \\ x_3 = (4 - x_1 + 2x_2)/3 \end{cases}$$

б) $x_1 = 0,98$

$x_2 = 0,000667$

$x_3 = 1,006$

Решите самостоятельно следующие задания:

Задание 2.1

Дана система линейных уравнений:

$$\begin{cases} x_1 + 5x_2 + x_3 = 11 \\ 4x_1 - x_2 + x_3 = 2 \\ x_1 + x_2 + 6x_3 = 3 \end{cases}$$

- а) Привести систему к виду, удобному для итераций, с учетом условий сходимости;
- б) Сделать три шага методом Зейделя, взяв $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$;
- с) Сравнить результаты с результатами задания 1.1.

✓ Задание 2.2

Дана система линейных уравнений:

$$\begin{cases} x_1 + 5x_2 + x_3 = 12 \\ 4x_1 - x_2 + x_3 = 3 \\ 2x_1 + x_2 + 8x_3 = 12 \end{cases}$$

- а) Привести систему к виду, удобному для итераций, с учетом условий сходимости;
- б) Сделать три шага методом Зейделя, взяв $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$;
- в) Сравнить результаты с результатами задания 1.2.

Задание 2.3

Дана система линейных уравнений:

$$\begin{cases} x_1 + 5x_2 + x_3 = 13 \\ 4x_1 - x_2 + x_3 = 4 \\ 3x_1 + x_2 + 10x_3 = 25 \end{cases}$$

- а) Привести систему к виду, удобному для итераций, с учетом условий сходимости;
- б) Сделать три шага методом Зейделя, взяв $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$;
- в) Сравнить результаты с результатами задания 1.3.

3. Пример выполнения упражнения тренинга на умение 3

Задание

Задана матрица A второго порядка:

$$A = \begin{bmatrix} 5 & 1 \\ 1 & 2 \end{bmatrix}$$

- а) Найдите численным методом обратную матрицу A^{-1} ;
- б) Проверьте правильность вычисления A^{-1} с помощью соотношения $A A^{-1} = E$.

Решение

1. Запишем расширенную матрицу, приписав справа к исходной матрице единичную матрицу:

$$\left[\begin{array}{cc|cc} 5 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{array} \right]$$

2. С помощью эквивалентных преобразований сведем исходную матрицу A к верхней треугольной матрице, причем преобразования будем выполнять

со всей "длинной" строкой. Из второй строки вычтем первую строку, умноженную на $1/5$. В результате расширенная матрица примет вид:

$$\left[\begin{array}{cc|cc} 5 & 1 & 1 & 0 \\ 0 & 9/5 & -1/5 & 1 \end{array} \right]$$

3. Для нахождения обратной матрицы надо решить две линейные системы уравнений с одинаковой матрицей, которая имеет после преобразования верхний треугольный вид, и двумя разными правыми частями:

$$\begin{cases} 5x_1 + x_2 = 1 \\ 1,8x_2 = -0,2 \end{cases} \quad \begin{cases} 5y_1 + y_2 = 0 \\ 1,8y_2 = 1 \end{cases}$$

В результате обратная матрица A^{-1} будет иметь вид:

$$A^{-1} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix}$$

Решим первую систему. Из второго уравнения найдем $x_2 = (-1/5)/(9/5) = -1/9$. Подставив x_2 в первое уравнение, найдем x_1 :

$$5x_1 = 1 - x_2 \Rightarrow x_1 = (1 - x_2)/5 = \frac{10}{9 \cdot 5} = \frac{2}{9}$$

Итак, решение первой системы: $x_1 = 2/9; x_2 = -1/9$. Решим вторую систему. Из второго уравнения $y_2 = 5/9$. Подставив y_2 в первое уравнение, найдем $y_1 = (-1/5), y_2 = -1/9$. Итак, решение второй системы $y_1 = -1/9, y_2 = 5/9$.

Записываем A^{-1} :

$$A^{-1} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix} = \begin{bmatrix} 2/9 & -1/9 \\ -1/9 & 5/9 \end{bmatrix}$$

4. Для контроля вычислений проверим выполнение равенства $A A^{-1} = E$.

$$A A^{-1} = \begin{bmatrix} 5 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 2/9 & -1/9 \\ -1/9 & 5/9 \end{bmatrix} =$$

$$= \begin{bmatrix} 5 \cdot 2/9 - 1 \cdot 1/9 & 5 \cdot (-1/9) + 1 \cdot 5/9 \\ 1 \cdot 2/9 - 2 \cdot 1/9 & 1 \cdot (-1/9) + 2 \cdot 5/9 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Значит обратная матрица A^{-1} найдена верно.

Ответ: $A^{-1} = \begin{bmatrix} 2/9 & -1/9 \\ -1/9 & 5/9 \end{bmatrix}$.

Решите самостоятельно следующие задания:

Задание 3.1

Задана матрица A второго порядка:

$$A = \begin{bmatrix} 4 & -1 \\ -1 & 1 \end{bmatrix}$$

- Найдите численным методом обратную матрицу A^{-1} ;
- Проверьте правильность вычисления A^{-1} с помощью соотношения $A A^{-1} = E$.

Задание 3.2

Задана матрица A второго порядка:

$$A = \begin{bmatrix} 5 & 0 \\ 0 & 2 \end{bmatrix}$$

- а) Найдите численным методом обратную матрицу A^{-1} ;
- б) Проверьте правильность вычисления A^{-1} с помощью соотношения $AA^{-1} = E$.

Задание 3.3

Задана матрица A второго порядка:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 8 \end{bmatrix}$$

- а) Найдите численным методом обратную матрицу A^{-1} ;
- б) Проверьте правильность вычисления A^{-1} с помощью соотношения $A A^{-1} = E$.

4. Пример выполнения упражнения тренинга на умение 4.

Задание

Задана матрица A второго порядка:

$$A = \begin{bmatrix} 5 & 1 \\ 1 & 2 \end{bmatrix}$$

- а) Степенным методом найдите максимальное по модулю собственное значение и соответствующий ему собственный вектор;
- б) Проверьте, с какой точностью полученный вектор будет собственным.

Решение

1. Зададим начальный вектор $\bar{y}_0 \{1,1\}$. Нормируем его, сделав его длину равной единице. Для этого вычислим длину вектора $\bar{y}_0$: $d_0 = \sqrt{1+1} = \sqrt{2}$. Разделим каждую компоненту вектора $\bar{y}_0$ на величину d_0 . Получим вектор $\bar{u}_0 \left\{ \frac{1}{\sqrt{2}}; \frac{1}{\sqrt{2}} \right\}$, длина которого равна 1.

2. Получим вектор первого приближения $\bar{y}_1$, умножив матрицу A на вектор $\bar{u}_0$:

$$\bar{y}_1 = \begin{bmatrix} 5 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix} = \begin{bmatrix} 5/\sqrt{2} & + & 1/\sqrt{2} \\ 1/\sqrt{2} & + & 2/\sqrt{2} \end{bmatrix} = \begin{bmatrix} 6/\sqrt{2} \\ 3/\sqrt{2} \end{bmatrix} = \begin{bmatrix} 4,2426 \\ 2,1213 \end{bmatrix}.$$

На этом заканчивается одна итерация степенного метода.

3. Проделаем теперь операции п.п. 1 и 2 с вектором $\bar{y}_1$:

$$d_1 = \sqrt{(4,2426)^2 + (2,1213)^2} = \sqrt{22,4996} = 4,7434$$

$$\bar{u}_1 = \{4,2426/4,7434; \quad 2,1213/4,7434\} = \{0,8944; \quad 0,4472\}$$

$$\bar{y}_2 = \begin{bmatrix} 5 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 0,8944 \\ 0,4472 \end{bmatrix} = \begin{bmatrix} 5 \cdot 0,8944 + 1 \cdot 0,4472 \\ 1 \cdot 0,8944 + 2 \cdot 0,4472 \end{bmatrix} = \begin{bmatrix} 4,9192 \\ 1,7888 \end{bmatrix}$$

Проделаем еще одну итерацию, выполнив п.п. 1 и 2 с вектором $\bar{y}_2$:

$$d_2 = \sqrt{(4,9192)^2 + (1,7888)^2} = 5,2343$$

$$\bar{u}_2 = \begin{bmatrix} 4,9192/5,2343 \\ 1,7888/5,2343 \end{bmatrix} = \begin{bmatrix} 0,9398 \\ 0,3417 \end{bmatrix}$$

$$\bar{y}_3 = \begin{bmatrix} 5 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 0,9398 \\ 0,3417 \end{bmatrix} = \begin{bmatrix} 5 \cdot 0,9398 + 1 \cdot 0,3417 \\ 1 \cdot 0,9398 + 2 \cdot 0,3417 \end{bmatrix} = \begin{bmatrix} 5,0007 \\ 1,6232 \end{bmatrix}$$

Вектор $\bar{y}_3$ можно приближенно считать собственным вектором, соответствующим максимальному собственному значению. Для удобства нормируем его, сделав его длину равной 1.

$$d_3 = \sqrt{(5,0007)^2 + (1,6232)^2} = 5,2575$$

$$\bar{u}_3 = \begin{bmatrix} 5,0007/5,2575 \\ 1,6232/5,2575 \end{bmatrix} = \begin{bmatrix} 0,9512 \\ 0,3087 \end{bmatrix}$$

Приближенно можно считать, что максимальное собственное значение $\lambda_{\max} \approx d_3$, а соответствующий собственный вектор $\bar{u}_{\max} \approx \bar{u}_3$.

4. Проверим, будет ли $\bar{u}_3$ собственным вектором. Проверим для этого выполнение соотношения:

$$A\bar{u}_3 = d_3\bar{u}_3$$

$$A\bar{u}_3 = \begin{bmatrix} 5 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 0,9512 \\ 0,3087 \end{bmatrix} = \begin{bmatrix} 5,0647 \\ 1,5686 \end{bmatrix} \quad d_3\bar{u}_3 = \begin{bmatrix} 5,0093 \\ 1,6230 \end{bmatrix}$$

По определению вектор $\bar{u}$ называется собственным для матрицы A, если он удовлетворяет соотношению $A\bar{u} = \lambda \bar{u}$. Видно, что это соотношение приближенно выполняется, так как $A\bar{u}_3 \approx d_3\bar{u}_3$. Для получения более высокой точности надо выполнить еще несколько итераций.

Ответ: $\lambda_{\max} \approx 5,2575$, $\bar{u}_{\max} \approx \{0,9512; 0,3087\}$.

Решите самостоятельно следующие задания:

Задание 4.1

Задана матрица A второго порядка:

$$A = \begin{bmatrix} 5 & 0 \\ 0 & 2 \end{bmatrix}$$

- а) Степенным методом найдите максимальное собственное значение и соответствующий ему собственный вектор;
- б) Проверьте, с какой точностью полученный вектор будет собственным.

Задание 4.2

Задана матрица A второго порядка:

$$A = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}$$

- a) Степенным методом найдите максимальное собственное значение и соответствующий ему собственный вектор;
- b) Проверьте, с какой точностью полученный вектор будет собственным.

Задание 4.3

Задана матрица A второго порядка:

$$A = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}$$

- a) Степенным методом найдите максимальное собственное значение и соответствующий ему собственный вектор;
- b) Проверьте, с какой точностью полученный вектор будет собственным.

5. Пример выполнения упражнения тренинга на умение 5

Задание

Дано нелинейное уравнение: $F(x) \equiv e^x + x^2 - 2 = 0$.

- a) Найти количество корней и отделить корни;
- b) Найти приближенное значение наибольшего корня методом половинного деления, сделав три шага;
- c) Оценить сходимость метода Ньютона и найти для него хорошее начальное приближение x_0 , используя результаты b);
- d) Найти двухстороннюю оценку для корня, сделав 1-2 шага комбинированным методом Ньютона и хорд.

Решение

1. Произведем отделение корней и определим их количество. Запишем уравнение $F(x) = 0$ в виде $e^x = 2 - x^2$. Построим графики функций, стоящих в левой и правой части (рис. 1).

Абсциссы точек пересечения графиков являются корнями уравнения. Видно, что уравнение имеет два корня. Положительный корень находится в интервале $[0; 1]$, а отрицательный – в интервале $[-\sqrt{2}; -1]$. Обозначим отрицательный корень через X_1 , а положительный – через X_2 . Наибольшим корнем будет X_2 .

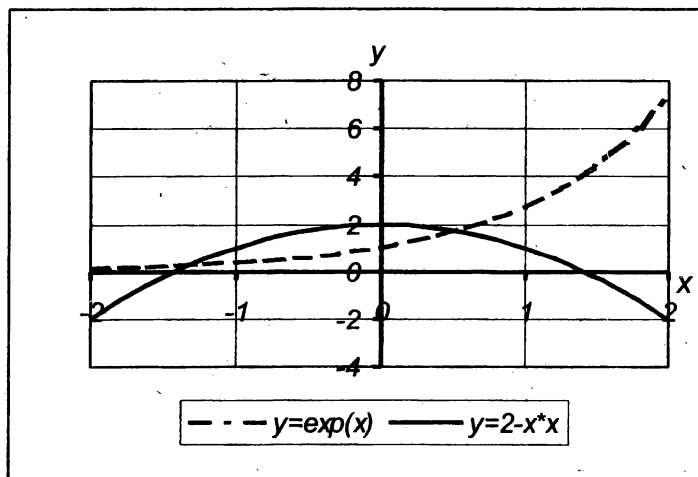


Рис. 1. Определение корней системы уравнений

2. Сделаем три шага методом половинного деления, чтобы уменьшить исходный интервал $[0; 1]$, на котором находится положительный корень X_2 . Метод изложен в разделе 5.3. В нашем случае $a = 0$, $b = 1$.

1 шаг. Вычислим середину отрезка $c = 0,5(a + b) = 0,5$ и величину $F(c) = e^{0,5} + (0,5)^2 - 2 = -0,1 < 0$. В исходном интервале $F(a) = F(0) = e^0 + 0^2 - 2 = -1 < 0$, а $F(b) = F(1) = e^1 + 1^2 - 2 = 1,7183 > 0$. На отрезке $[0; 0,5]$ на концах функция имеет одинаковые знаки ($F(0) < 0$, $F(0,5) < 0$) и значит корня там нет, следовательно, $X_2 \in [0,5; 1]$. Примем за новый интервал $a_1 = 0,5$ и $b_1 = 1$.

2 шаг. Вычислим $c_1 = 0,5(a_1 + b_1) = 0,5(0,5 + 1) = 0,75$ и определим знак функции $F(x)$ в точке c_1 . Получим $F(c_1) = F(0,75) = e^{0,75} + (0,75)^2 - 2 = 0,6795 > 0$. Видно, что $F(a_1) \cdot F(c_1) = F(0,5) \cdot F(0,75) = (-0,1) \cdot 0,6795 < 0$. Следовательно, $X_2 \in [0,5; 0,75]$. Примем $a_2 = 0,5$ и $b_2 = 0,75$.

3 шаг. $c_2 = 0,5(a_2 + b_2) = 0,5(0,5 + 0,75) = 0,625$. Вычислим знак $F(c_2)$: $F(c_2) = F(0,625) = e^{0,625} + (0,625)^2 - 2 = 0,2589 > 0$. Следовательно, $F(a_2) \cdot F(c_2) = F(0,5) \cdot F(0,625) = (-1) \cdot 0,2589 < 0$. Отсюда следует, что $X_2 \in [0,5; 0,625]$. Итак, методом половинного деления мы уменьшили исходный интервал $[0; 1]$ в 8 раз и получили оценку для корня: $0,5 < X_2 < 0,625$.

Представим полученные результаты в виде таблицы.

Таблица 1

a_k	b_k	c_k	$F(a_k)$	$F(b_k)$	$F(c_k)$	
0	1	0,5	-1	1,7183	-0,1013	$x_2^* \in [0,5; 1]$
0,5	1	0,75	-0,1	1,7183	0,6795	$x_2^* \in [0,5; 0,75]$
0,5	0,75	0,625	-0,1	0,6795	0,2589	$x_2^* \in [0,5; 0,625]$

3. Оценим сходимость метода Ньютона на интервале $[0,5; 0,625]$ (см. раздел 5.5). Для этого вычислим первую и вторую производную функции $F(x)$ и оценим их знаки на интервале. $F(x) = e^x + 2x$, $F'(x) = e^x + 2$. Видно, что для любых положительных значений x $F'(x) > 0$ и $F''(x) > 0$ и значит эти функции знакопостоянные. Найдем x_0 , для которого сходимость метода Ньютона гарантирована. Для этого должно быть $F(x_0)F''(x_0) > 0$. Вычислим $F(0,625) \cdot F''(0,625) = 0,2589 \cdot 3,8682 > 0$. Значит, надо взять $x_0 = 0,625$. Вычислим первое приближение по методу Ньютона:

$$x_1 = x_0 - F(x_0)/F'(x_0) = 0,625 - F(0,625)/F'(0,625) = \\ = 0,625 - (e^{0,625} + (0,625)^2 - 2) / (e^{0,625} + 2 \cdot 0,625) = 0,5420.$$

Приняв теперь за исходное приближение $x_1 = 0,5420$, вычислим второе приближение метода Ньютона:

$$x_2 = x_1 - F(x_1)/F'(x_1) = 0,5420 - F(0,5420)/F'(0,5420) = \\ = 0,5420 - (e^{0,5420} + (0,5420)^2 - 2) / (e^{0,5420} + 2 \cdot 0,5420) = 0,5373.$$

4. Сделаем один шаг методом хорд, приняв за исходный интервал $[0,5; 0,625]$: $a_0 = 0,5$ и $b_0 = 0,625$. Используя формулу метода хорд, получим:

$$b_1 = b_0 - F(b_0)(a_0 - b_0) / (F(a_0) - F(b_0)) = \\ = 0,625 - 0,2589(-0,125) / (-0,1013 - 0,2589) = 0,5352.$$

Так как на исходном отрезке выполняются условия Фурье ($F'(x)$ и $F''(x)$ – знакопостоянные функции, $F(x_0)F''(x_0) > 0$ для $x_0 = 0,625$), то метод хорд и метод Ньютона сходятся монотонно. При этом метод хорд дает приближения с недостатком, а метод Ньютона – с избытком. Отсюда следует, что для X_2 существует оценка $0,5352 < X_2 < 0,5373$.

Ответ: положительный корень уравнения находится в интервале $[0,5352; 0,5373]$.

Решите самостоятельно следующие задания:

Задание 5.1

Задано нелинейное уравнение: $2\sin x + x - 1 = 0$.

- а) Отделить корни уравнения;
- б) Найти приближенное значение корня методом половинного деления, сделав три шага;
- с) Оценить сходимость метода Ньютона и найти для него хорошее начальное приближение x_0 , используя результаты б);
- д) Найти оценку для корня, сделав по одному шагу комбинированным методом Ньютона и хорд.

Задание 5.2

Дано нелинейное уравнение: $e^x + x^2 - 1 = 0$.

- a) Найти количество корней и отделить корни;
- b) Найти приближенное значение наибольшего корня методом половинного деления, сделав три шага;
- c) Оценить сходимость метода Ньютона и найти для него хорошее начальное приближение x_0 , используя результаты b;
- d) Найти двухстороннюю оценку для корня, сделав 1-2 шага комбинированным методом Ньютона и хорд.

Задание 5.3

Дано нелинейное уравнение: $\cos x - x^2 = 0$.

- a) Найти количество корней и отделить корни;
- b) Найти приближенное значение наибольшего корня методом половинного деления, сделав три шага;
- c) Оценить сходимость метода Ньютона и найти для него хорошее начальное приближение x_0 , используя результаты b);
- d) Найти двухстороннюю оценку для корня, сделав 1-2 шага комбинированным методом Ньютона и хорд.

6. Пример выполнения упражнения тренинга на умение 6

Задание

Задана система двух нелинейных уравнений с двумя неизвестными:

$$\begin{cases} F_1(x, y) \equiv y(x - 1) - 1 = 0 \\ F_2(x, y) \equiv x^2 - y^2 - 1 = 0 \end{cases}$$

- Проанализировать систему, найдя количество корней и их расположение;
- Получить первое приближение методом Ньютона для корня уравнения, лежащего в третьей четверти.

Решение

1. Оба уравнения представляют собой гиперболы, вид которых приведен на рис. 6.1 главы 6. В третьей четверти находится корень, для которого $x \approx -1, y \approx -0,5$.

2. Примем начальные значения для метода Ньютона $x_0 = -1, y_0 = -0,5$.

Вычислим частные производные, входящие в Якобиан системы: $\frac{\partial F_1}{\partial x} = y$;

$$\frac{\partial F_1}{\partial y} = x - 1; \quad \frac{\partial F_2}{\partial x} = 2x; \quad \frac{\partial F_2}{\partial y} = -2y. \text{ Подставив начальное приближение } x_0 = -1,$$

$y_0 = -0,5$ в полученные функции, определяем численное значение элементов

якобиана $\left(\frac{\partial F_1}{\partial x}\right)_0 = -0,5; \left(\frac{\partial F_1}{\partial y}\right)_0 = -2; \left(\frac{\partial F_2}{\partial x}\right)_0 = -2; \left(\frac{\partial F_2}{\partial y}\right)_0 = 1$. Для

получения поправок Δx_0 и Δy_0 получаем линейную систему уравнений:

$$\begin{cases} -0,5\Delta x_0 - 2\Delta y_0 = -F_1(x_0, y_0) = 0 \\ -2\Delta x_0 + \Delta y_0 = -F_2(x_0, y_0) = -0,25 \end{cases}$$

Решим эту систему методом Гаусса. Из второго уравнения вычтем первое уравнение, умноженное на 4. Получим систему с верхней треугольной матрицей:

$$\begin{cases} -0,5\Delta x_0 - 2\Delta y_0 = 0 \\ 9\Delta y_0 = 0,25 \end{cases}$$

Обратным ходом метода Гаусса найдем решение системы $\Delta y^{(0)} = 0,02778$,

$\Delta x^{(0)} = -0,1111$. Определяем первое приближение: $x_1 = x_0 + \Delta x_0 = -1 - 0,1111 = -1,1111$; $y_1 = y_0 + \Delta y_0 = -0,5 + 0,02778 = -0,4772$. На этом один шаг метода Ньютона заканчивается.

Ответ:

- а) у системы два корня, один лежит в первой четверти, а второй – в третьей четверти;
б) $x_1 = -1,1111$; $y_1 = -0,4772$.

Решите самостоятельно следующие задания:

✓ **Задание 6.1**

Задана система двух нелинейных уравнений с двумя неизвестными:

$$\begin{cases} y - x^2 + 0,2 = 0 \\ x - y^2 = 0 \end{cases}$$



- а) Проанализировать систему, найдя количество корней и их расположение;
б) Получить первое приближение методом Ньютона для корня уравнения, лежащего в первой четверти, взяв $x_0 = y_0 = 1$.

Задание 6.2

Задана система двух нелинейных уравнений с двумя неизвестными:

$$\begin{cases} y - \ln x = 0 \\ x^2 + y^2 - 3 = 0 \end{cases}$$

- Проанализировать систему, найдя количество корней и их расположение;
- Получить первое приближение методом Ньютона для корня уравнения, лежащего в первой четверти, взяв $x_0 = y_0 = 2$.

Задание 6.3

Задана система двух нелинейных уравнений с двумя неизвестными:

$$\begin{cases} y^2 + x^2 - 3 = 0 \\ x - y^2 = 0 \end{cases}$$

- a) Проанализировать систему, найдя количество корней и их расположение;
- b) Получить первое приближение методом Ньютона для корня уравнения, лежащего в первой четверти, взяв $x_0 = y_0 = 1,5$.

ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Составьте логическую схему базы знаний по теме юниты.

2. Решите самостоятельно следующие задачи, выбрав свой вариант. Во все задачи входит величина S – номер студента в журнале. Задав величину S , студент определяет условия задачи своего варианта.

1) Задана система линейных уравнений

$$\begin{cases} x_1 + 5x_2 + x_3 = S + 6 \\ 4 \cdot x_1 - x_2 + x_3 = S - 12 \\ S \cdot x_1 + 2x_2 + 2 \cdot S \cdot x_3 = 2 \cdot S^2 - 6 \cdot S + 4 \end{cases}$$

- а) Решить систему методом Гаусса.
- б) Найти LU -разложение матрицы A исходной системы.
- в) С помощью LU -разложения найти обратную матрицу A^{-1} .
- г) Записать систему в виде, удобном для итераций, с учетом условий сходимости.
- д) Сделать три итерации методом Зейделя.

2) Задано нелинейное уравнение

$$x^3 + x - a = 0$$

Величина a определяется по номеру студента в журнале

$$a = \frac{S^3 + S \cdot (S + 10)^2}{(S + 10)^3}$$

- а) Определить интервал, в котором находится корень уравнения.
- б) Сузить этот интервал, сделав 3–4 шага методом половинного деления или хорд.
- в) Уточнить значение корня, сделав 1–2 шага комбинированным методом хорд и Ньютона.

